**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN TỐT NGHIỆP**

**NGHIÊN CỨU SPRING BOOT VÀ SPRING SECURITY  
XÂY DỰNG ỨNG DỤNG QUẢN LÝ   
CHUỖI CỬA HÀNG TOUS LES TEMP**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn** | **: ThS. Mai Cường Thọ** |
| **Sinh viên thực hiện** | **: Nguyễn Lê Thành Tâm** |
| **Mã số sinh viên** | **: 61134311** |

|  |
| --- |
| Khánh Hòa, tháng 6 – 2023 |

|  |
| --- |
| **TRƯỜNG ĐẠI HỌC NHA TRANG**  **KHOA CÔNG NGHỆ THÔNG TIN**  **BỘ MÔN HỆ THỐNG THÔNG TIN** |

****

**ĐỒ ÁN TỐT NGHIỆP**

**NGHIÊN CỨU SPRING BOOT VÀ SPRING SECURITY  
XÂY DỰNG ỨNG DỤNG QUẢN LÝ   
CHUỖI CỬA HÀNG TOUS LES TEMP**

|  |  |
| --- | --- |
| GVHD | : ThS. Mai Cường Thọ |
| SVTH | : Nguyễn Lê Thành Tâm |
| MSSV | : 61134311 |

|  |
| --- |
| Khánh Hòa, tháng 6 – 2023 |

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ ĐỒ ÁN TỐT NGHIỆP**

***(Dùng cho CBHD và nộp cùng báo cáo Đồ án của sinh viên)***

**Tên đề tài:** NGHIÊN CỨU SPRING BOOT VÀ SPRING SECURITY XÂY DỰNG ỨNG DỤNG QUẢN LÝ CHUỖI CỬA HÀNG TOUS LES TEMP

**Giảng viên hướng dẫn:** ThS. Mai Cường Thọ

**Sinh viên được hướng dẫn:** Nguyễn Lê Thành Tâm **MSSV:** 61134311

**Khóa:** 61 **Ngành:** Công nghệ thông tin

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Lần KT** | **Ngày** | | **Nội dung** | | | | **Nhận xét của GVHD** |
| 1 | 10/03 | | Trao đổi đề tài | | | |  |
| 2 | 20/03 | | Viết đề cương đề tài | | | |  |
| 3 | 26/03 | | Khảo sát hệ thống chuỗi cửa hàng  Tous Les Temp | | | |  |
| 4 | 10/04 | | Vẽ sơ đồ phân rã chức năng | | | |  |
| 5 | 22/04 | | Vẽ sơ đồ ERD | | | |  |
| 6 | 05/05 | | Thiết kế CSDL | | | |  |
| 7 | 12/05 | | Thiết kế và cài đặt hệ thống | | | |  |
| **Kiểm tra giữa tiến độ của Trưởng BM** | | | | | | | |
| Ngày kiểm tra:  ………………….. | | Đánh giá công việc hoàn thành:………% | | | | Ký tên  …………………... | |
| Được tiếp tục: ☐ | | Không tiếp tục: ☐ | |
| 8 | |  | | |  |  | |
| 9 | |  | | |  |  | |
| 10 | |  | | |  |  | |
| 11 | |  | | |  |  | |
| 12 | |  | | |  |  | |
| 13 | |  | | |  |  | |
| 14 | |  | |  | |  | |
| 15 | |  | |  | |  | |

**Nhận xét chung** (sau khi sinh viên hoàn thành ĐA/KLTN):

|  |  |  |
| --- | --- | --- |
| Điểm hình thức:………/10 | Điểm nội dung:………/10 | **Điểm tổng kết**:………../10 |

|  |  |  |  |
| --- | --- | --- | --- |
| Đồng ý cho sinh viên: | | Được bảo vệ: ☐ | Không được bảo vệ: ☐ |
|  | *Khánh Hòa, ngày……tháng……năm…..* | | |
|  | **Giáo viên hướng dẫn**  *(Ký và ghi rõ họ tên)* | | |

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**PHIẾU CHẤM ĐIỂM ĐỒ ÁN TỐT NGHIỆP**

**(Dùng cho CBHD và nộp cùng báo cáo ĐA/KLTN của sinh viên)**

Tên đề tài: NGHIÊN CỨU SPRINGBOOT VÀ SPRING SECURITY XÂY DỰNG ỨNG DỤNG QUẢN LÝ CHUỖI CỬA HÀNG TOUS LES TEMP

Chuyên ngành: Công nghệ phần mềm

Họ và tên sinh viên: Nguyễn Lê Thành Tâm MSSV: 61134311

Người phản biện (học hàm, học vị, họ và tên): ThS. Mai Cuòng Thọ

Cơ quan công tác: Trường Đại học Nha Trang

**Phần đánh giá và cho điểm của người hướng dẫn (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí đánh giá** | **Trọng số (%)** | **Mô tả chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| **Xây dựng đề cương nghiên cứu** | **10** |  |  |  |  |  |
| **Tinh thần và thái độ làm việc** | **10** |  |  |  |  |  |
| **Kiến thức và kỹ năng làm việc** | **10** |  |  |  |  |  |
| **Nội dung và kết quả đạt được** | **40** |  |  |  |  |  |
| **Kỹ năng viết và trình bày báo cáo** | **30** |  |  |  |  |  |
| **ĐIỂM TỔNG** | | | | | |  |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

**Nhận xét chung** (sau khi sinh viên hoàn thành ĐA/KLTN):

|  |  |  |  |
| --- | --- | --- | --- |
| Đồng ý cho sinh viên: | | Được bảo vệ: ☐ | Không được bảo vệ: ☐ |
|  | *Khánh Hòa, ngày……tháng……năm…..* | | |
|  | **Cán bộ hướng dẫn**  *(Ký và ghi rõ họ tên)* | | |

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**PHIẾU CHẤM ĐIỂM ĐỒ ÁN TỐT NGHIỆP**

**(Dùng cho CBPB và nộp cùng báo cáo ĐA/KLTN của sinh viên)**

Tên đề tài: NGHIÊN CỨU SPRINGBOOT VÀ SPRING SECURITY XÂY DỰNG ỨNG DỤNG QUẢN LÝ CHUỖI CỬA HÀNG TOUS LES TEMP

Chuyên ngành: Công nghệ phần mềm

Họ và tên sinh viên: Nguyễn Lê Thành Tâm MSSV: 61134311

Người phản biện (học hàm, học vị, họ và tên): ThS. Mai Cường Thọ

Cơ quan công tác: Trường Đại học Nha Trang

**Phần đánh giá và cho điểm của người hướng dẫn (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí đánh giá** | **Trọng số (%)** | **Mô tả chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| **Xây dựng đề cương nghiên cứu** | **10** |  |  |  |  |  |
| **Tinh thần và thái độ làm việc** | **10** |  |  |  |  |  |
| **Kiến thức và kỹ năng làm việc** | **10** |  |  |  |  |  |
| **Nội dung và kết quả đạt được** | **40** |  |  |  |  |  |
| **Kỹ năng viết và trình bày báo cáo** | **30** |  |  |  |  |  |
| **ĐIỂM TỔNG** | | | | | |  |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

**Nhận xét chung:**

|  |  |  |  |
| --- | --- | --- | --- |
| Đồng ý cho sinh viên: | | Được bảo vệ: ☒ | Không được bảo vệ: ☐ |
|  | *Khánh Hòa, ngày……tháng……năm…..* | | |
|  | **Cán bộ phản biện**  *(Ký và ghi rõ họ tên)* | | |

**II. Phần nhận xét cụ thể (dựa theo phiếu chấm điểm và khung tiêu chí đánh giá theo Rubric)**

**II.1. Hình thức thuyết minh (tỉ trọng 30%)**

***\* Trình bày*** (*Rõ ràng, mạch lạc? Biểu bảng, hình vẽ trình bày rõ ràng, đúng quy cách?...)*

***\* Bố cục và lập luận*** *(Bố cục hợp lý? Tỉ trọng giữa các phần? Cơ sở lập luận?...)*

***\* Văn phong*** *(Gọn gàng, súc tích hay rườm rà, khó hiểu? Lỗi văn phạm và chính tả?…)*

**II.2. Nội dung thuyết minh** (tỉ trọng 30%)

***\* Mục tiêu nghiên cứu*** *(Trình bày rõ ràng? Ý nghĩa khoa học và thực tiễn? Tính khả thi?...)*

***\* Tổng quan tài liệu*** *(Phân tích và đánh giá? Độ tin cậy và chất lượng nguồn tài liệu?…)*

***\* Phương pháp nghiên cứu* (***Hiện đại?**Phù hợp với mục tiêu và nội dung nghiên cứu? Mô tả? Đánh giá và so sánh với các phương pháp khác?…)*

**II.3. Kết quả nghiên cứu (**tỉ trọng 20%)

***\* Kết quả đạt được*** *(Độ**tin cậy? Tính sáng tạo? Giá trị khoa học và thực tiễn?...)*

***\* Kết luận*** *(Đáp ứng mục tiêu nghiên cứu? Quan điểm của cá nhân?....)*

**II.4. MỨC ĐỘ TRÍCH DẪN VÀ SAO CHÉP** (tỉ trọng 20%)

**\* Mức độ trích dẫn *(****Đúng quy định? Trung thực, đầy đủ, rõ ràng? Sắp xếp tài liệu tham khảo?...)*

**\* Mức độ sao chép** *(Tỉ lệ sao chép? Hình thức sao chép?...)*

**KẾT QUẢ KIỂM TRA ĐẠO VĂN BẰNG TURNITIN**

LỜI CAM ĐOAN

Em xin cam đoan đồ án “Nghiên cứu Springboot và Spring Sercurity xây dựng ứng dụng quản lý chuỗi cửa hàng TOUS LES TEMP” là kết quả nghiên cứu và nỗ lực của bản thân em.

|  |
| --- |
| Nha Trang, ngày 03 tháng 05 năm 2023 |
| Sinh viên cam đoan |
| **Nguyễn Lê Thành Tâm** |

LỜI CÁM ƠN

Trong suốt khoảng thời gian tại Trường Đại học Nha Trang, đối với em nó là khoảng thời gian vô cùng quý báu và bổ ích. Được tiếp thu những kiến thức từ những thầy, cô trong trường cũng như được đúc kết kinh nghiệm trong quá trình thực tập thực tế tại môi trường công ty để có thể vận dụng vào đồ án tốt nghiệp của mình và cho công việc sau này. Trong thời gian hoàn thành đồ án, ngoài sự nỗ lực của bản thân thì em còn nhận được rất nhiều sự giúp đỡ, động viên tận tình từ phía thầy cô, gia đình và bạn bè.

Em xin chân thành cảm ơn đến toàn thể thầy, cô đang công tác tại Trường Đại học Nha Trang nói chung các và thầy, cô công tác tại Khoa Công nghệ thông tin nói riêng, những người đã dìu dắt, trau dồi những kiến thức bổ ích cho em trong suốt bốn năm đại học này.

Đặc biệt, em xin chân thành cám ơn sâu sắc đến cô giáo ThS. Mai Cường Thọ đã trực tiếp hướng dẫn, tận tình chỉ bảo, tạo mọi điều kiện giúp đỡ em trong suốt quá trình làm đồ án tốt nghiệp lần này.

Ngoài ra, em cũng xin gửi lời cám ơn của mình đến toàn thể các anh (chị) tại Công ty FPT Software Quy Nhơn đã hỗ trợ, giúp đỡ em trong quá trình em thực tập tại đơn vị.

Vì thời gian, năng lực có hạn và còn nhiều hạn chế nên đồ án của em không thể tránh khỏi nhiều điều sai sót. Em rất mong nhận được sự góp ý của các thầy, cô để có thể hoàn thành đồ án tốt hơn và có thêm kinh nghiệm để phục vụ cho công việc sau này.

|  |
| --- |
| Nha Trang, ngày 03 tháng 05 năm 2022 |
| Sinh viên thực hiện |
| **Nguyễn Lê Thành Tâm** |

TÓM TẮT ĐỒ ÁN TỐT NGHIỆP

Đề tài “Nghiên cứu Springboot và Spring Sercurity xây dựng ứng dụng quản lý chuỗi cửa hàng Tous Les Temp” tập trung vào việc tìm hiểu về Spring Boot và Spring Security, các tính năng, ưu điểm, cách sử dụng và các thành phần chính của công nghệ này. Ứng dụng sẽ có các tính năng quản lý nhân viên, quản lý sản phẩm từng cửa hàng, quản lý lương nhân viên, quản lý địa chỉ của chuỗi cửa hàng thông qua google map, thống kê lương cho nhân viên,…

Hiện tại, hệ thống vẫn còn tồn tại một số hạn chế nhất định nên một số chức năng vẫn chưa được hoàn thiện. Do đó, em chỉ xin phép được trình bày các chức năng thực hiện được như quản lí nhân viên, sản phẩm, loại sản phẩm, danh sách cửa hàng thông qua google map, thống kê lương nhân viên, cùng với đó là thiết kế sơ đồ luồng dữ liệu, cơ sở dữ liệu và giao diện cho hệ thống. Nêu ra được ưu và nhược điểm của hệ thống đang gặp phải. Em mong nhận được sự cảm thông của thầy cô và các bạn.

MỤC LỤC

[LỜI CAM ĐOAN i](#_Toc136900486)

[LỜI CÁM ƠN ii](#_Toc136900487)

[TÓM TẮT ĐỒ ÁN TỐT NGHIỆP iii](#_Toc136900488)

[MỤC LỤC iv](#_Toc136900489)

[DANH MỤC HÌNH ẢNH vii](#_Toc136900490)

[DANH MỤC BẢNG ix](#_Toc136900491)

[DANH MỤC TỪ VIẾT TẮT ix](#_Toc136900492)

[PHẦN MỞ ĐẦU x](#_Toc136900493)

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN BÀI TOÁN VÀ CƠ SỞ LÝ THUYẾT 1](#_Toc136900494)

[1.1 Giới thiệu tổng quan về chuỗi cửa hàng tOus Les temps 1](#_Toc136900495)

[1.1.1 Một số nét về chuỗi cửa hàng Tous Les Temp 1](#_Toc136900496)

[1.1.2 Tổng quan hiện trạng tại chuỗi cửa hàng Tous Les Temps 1](#_Toc136900497)

[1.1.3 Giải pháp đề xuất cho bài toán quản lý chuỗi cửa hàng 1](#_Toc136900498)

[1.2 Cơ sở lý thuyết 2](#_Toc136900499)

[1.2.1 Một số khái niệm cơ bản 2](#_Toc136900500)

[1.2.2 Quy trình phát triển hệ thống thông tin 3](#_Toc136900501)

[1.2.3 Mô hình hóa hệ thống thông tin 4](#_Toc136900502)

[1.3 Phương pháp nghiên cứu 4](#_Toc136900503)

[1.3.1 Phương pháp tìm kiếm, khảo sát thông tin 4](#_Toc136900504)

[1.3.2 Các phương pháp phân tích, thiết kế hệ thống 5](#_Toc136900505)

[1.4 Môi trường lập trình 6](#_Toc136900506)

[1.4.1 IntelliJ IDEA 6](#_Toc136900507)

[1.4.2 Microsoft Visual Studio Code 7](#_Toc136900508)

[1.4.3 Postman 7](#_Toc136900509)

[1.4.4 PowerDesigner 7](#_Toc136900510)

[1.5 Công nghệ sử dụng 7](#_Toc136900511)

[1.5.1 Spring Boot Framework 7](#_Toc136900512)

[1.5.2 Spring Security Framework 9](#_Toc136900513)

[1.5.3 Angular 10](#_Toc136900514)

[1.5.4 Two Factor Authentication 10](#_Toc136900515)

[1.5.5 Google Map API 12](#_Toc136900516)

[1.5.6 Swagger 13](#_Toc136900517)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG CHO BÀI TOÁN QUẢN LÝ CHUỖI CỬA HÀNG tOUS LES TEMP 15](#_Toc136900518)

[2.1 Đặc tả hệ thống 15](#_Toc136900519)

[2.1.1 Quản lý danh sách cửa hàng 15](#_Toc136900520)

[2.1.2 Quản lý sản phẩm 16](#_Toc136900521)

[2.1.3 Quản lý nhân viên 17](#_Toc136900522)

[2.1.4 Tra cứu thông tin 17](#_Toc136900523)

[2.1.5 Chấm công nhân viên 18](#_Toc136900524)

[2.1.6 Báo cáo thống kê 19](#_Toc136900525)

[2.2 Sơ đồ mô tả hệ thống 20](#_Toc136900526)

[2.2.1 Sơ đồ phân rã chức năng 20](#_Toc136900527)

[2.2.2 Sơ đồ phân rã mức ngữ cảnh 22](#_Toc136900528)

[2.2.3 Sơ đồ phân rã mức 0 22](#_Toc136900529)

[2.2.4 Sơ đồ phân rã mức 1 Quản lý danh sách cửa hàng 23](#_Toc136900530)

[2.2.5 Sơ đồ phân rã mức 1 Quản lý danh sách sản phẩm 23](#_Toc136900531)

[2.2.6 Sơ đồ phân rã mức 1 Quản lý nhân viên 24](#_Toc136900532)

[2.2.7 Sơ đồ phân rã mức 1 Tra cứu 24](#_Toc136900533)

[2.2.8 Sơ đồ phân rã mức 1 Chấm công nhân viên 25](#_Toc136900534)

[2.2.9 Sơ đồ phân rã mức 1 Báo cáo thống kê 25](#_Toc136900535)

[2.3 Đánh giá hệ thống dữ liệu 26](#_Toc136900536)

[2.3.1 Phân tích hệ thống quản lý chuỗi cửa hàng Tous Les Temp 26](#_Toc136900537)

[2.3.2 Các thực thể liên kết 27](#_Toc136900538)

[2.3.3 Biến đổi sơ đồ thực thể liên kết sang sơ đồ dữ liệu quan hệ 29](#_Toc136900539)

[2.4 Cơ sở dữ liệu 30](#_Toc136900540)

[CHƯƠNG 3: LẬP TRÌNH ỨNG DỤNG 36](#_Toc136900541)

[3.1 Cấu trúc hệ thống 36](#_Toc136900542)

[3.2 Các giao diện trong hệ thống 39](#_Toc136900543)

[3.2.1 Giao diện đăng ký, đăng nhập và xác thực 2FA 39](#_Toc136900544)

[3.2.2 Thanh menu hệ thống 44](#_Toc136900545)

[3.2.3 Giao diện quản lý cửa hàng 45](#_Toc136900546)

[3.2.4 Giao diện quản lý sản phẩm 48](#_Toc136900547)

[3.2.5 Giao diện quản lý nhân viên 52](#_Toc136900548)

[3.2.6 Giao diện quản lý chấm công 55](#_Toc136900549)

[3.3 Một số xử lý tiêu biểu 59](#_Toc136900550)

[3.3.1 Spring Boot 59](#_Toc136900551)

[3.3.2 Angular 63](#_Toc136900552)

[KẾT LUẬN 66](#_Toc136900553)

[TÀI LIỆU THAM KHẢO 68](#_Toc136900554)

DANH MỤC HÌNH ẢNH

[Hình 1. 1 Cơ chế hoạt động của 2FA 11](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900567)

[Hình 1. 3 Swagger dependency 13](#_Toc136900568)

[Hình 1. 4 Class SwaggerConfig.java 14](#_Toc136900569)

[HInh 2. 1 Sơ đồ phân rã chức năng - FDD 20](#_Toc136900555)

[HInh 2. 2 Sơ đồ mức ngữ cảnh quản lý chuỗi cửa hàng 22](#_Toc136900556)

[HInh 2. 3 Sơ đồ phân rã DFD mức 0 22](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900557)

[HInh 2. 4 Sơ đồ phân rã DFD mức 1 Quản lý danh sách cửa hàng 23](#_Toc136900558)

[HInh 2. 5 Sơ đồ phân rã DFD mức 1 Quản lý sản phẩm 23](#_Toc136900559)

[HInh 2. 6 Sơ đồ phân rã DFD mức 1 Quản lý nhân viên 24](#_Toc136900560)

[HInh 2. 7 Sơ đồ phân rã DFD mức 1 Tra cứu 24](#_Toc136900561)

[HInh 2. 8 Sơ đồ phân rã DFD mức 1 Chấm công nhân viên 25](#_Toc136900562)

[HInh 2. 9 Sơ đồ phân rã DFD mức 1 Báo cáo thống kê 25](#_Toc136900563)

[HInh 2. 10 Sơ đồ dữ liệu quan hệ (ERD mức logic) 29](#_Toc136900564)

[HInh 2. 11 Sơ đồ dữ liệu quan hệ (ERD mức vật lý) 30](#_Toc136900565)

[HInh 2. 12 Lược đồ CSDL trên MySQL - Database Diagram 35](#_Toc136900566)

[Hình 3. 1 Mô hình kết hợp Spring và Angular 36](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900570)

[Hình 3. 2 Luồng dữ liệu phía backend 37](#_Toc136900571)

[Hình 3. 3 Cấu trúc source file 38](#_Toc136900572)

[Hình 3. 4 Đăng nhập hệ thống 39](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900573)

[Hình 3. 5 Xác thực OTP 40](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900574)

[Hình 3. 6 Đăng ký 41](#_Toc136900575)

[Hình 3. 7 Đăng ký thành công có sử dụng 2FA 42](#_Toc136900576)

[Hình 3. 8 Đăng ký thành công không sử dụng 2FA 43](#_Toc136900577)

[Hình 3. 9 Thông tin tài khoản 43](#_Toc136900578)

[Hình 3. 10 Dashboard của hệ thống 44](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900579)

[Hình 3. 11 Danh sách cửa hàng 45](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900580)

[Hình 3. 12 Cập nhật thông tin cửa hàng 46](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900581)

[Hình 3. 13 Thêm mới cửa hàng 46](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900582)

[Hình 3. 14 Đỉa chỉ các cửa hàng thông qua Google Map API 47](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900583)

[Hình 3. 15 Danh sách tấc cả sản phẩm 48](#_Toc136900584)

[Hình 3. 16 Tạo mới sản phẩm 49](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900585)

[Hình 3. 17 Cập nhật thông tin sản phẩm 50](#_Toc136900586)

[Hình 3. 18 Danh mục loại sản phẩm 50](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900587)

[Hình 3. 19 Thêm mới loại sản phẩm 51](#_Toc136900588)

[Hình 3. 20 Cập nhật loại sản phẩm 51](#_Toc136900589)

[Hình 3. 21 Quản lý danh sách nhân viên 52](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900590)

[Hình 3. 22 Dialog hiển thị ảnh nhân viên 52](#_Toc136900591)

[Hình 3. 23 Thêm mới nhân viên 53](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900592)

[Hình 3. 24 Cập nhật thông tin nhân viên 53](#_Toc136900593)

[Hình 3. 25 Danh sách nhóm nhân viên 54](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900594)

[Hình 3. 27 Danh sách lương cơ bản 55](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900595)

[Hình 3. 27 Tạo mới mức lương 55](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900596)

[Hình 3. 28 Cập nhật thông tin lương 56](#_Toc136900597)

[Hình 3. 29 Tạo bảng chấm công 56](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900598)

[Hình 3. 30 Quản lí danh sách bảng công 57](#_Toc136900599)

[Hình 3. 31 Cập nhật bảng chấm công 57](#_Toc136900600)

[Hình 3. 32 Bảng công nhân viên 58](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900601)

[Hình 3. 33 Thống kê lương nhân viên 58](#_Toc136900602)

[Hình 3. 34 Cấu hình Swagger cho việc quản lý tài liệu RESTful API 59](#_Toc136900603)

[Hình 3. 35 Giao diện Swagger 59](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900604)

[Hình 3. 36 Cấu hình Thymeleaf cho mail template 60](#_Toc136900605)

[Hình 3. 37 Class MailServiceImpl 60](#_Toc136900606)

[Hình 3. 38 Mail xác thực đăng ký tài khoản thành công 61](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900607)

[Hình 3. 39 Security Config 61](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900608)

[Hình 3. 40 File cấu hình của project application.yaml 62](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900609)

[Hình 3. 41 Class auth.interceptor.ts 64](#_Toc136900610)

[Hình 3. 42 Class home.guard.ts 64](#_Toc136900611)

[Hình 3. 43 Firebase config 65](file:///D:\DATN\report\Đồ%20án%20tốt%20nghiệp.docx#_Toc136900612)

DANH MỤC BẢNG

[Bảng 2. 1 Các thực thể, định danh và thuộc tính mô tả 27](#_Toc136894978)

[Bảng 2. 2 Bảng ADDRESS 30](#_Toc136894979)

[Bảng 2. 3 Bảng CATEGORY 30](#_Toc136894980)

[Bảng 2. 4 Bảng SIZE 31](#_Toc136894981)

[Bảng 2. 5 Bảng PRODUCT 31](#_Toc136894982)

[Bảng 2. 6 Bảng ROLE 31](#_Toc136894983)

[Bảng 2. 7 Bảng USER\_ROLE 32](#_Toc136894984)

[Bảng 2. 8 Bảng USER 32](#_Toc136894985)

[Bảng 2. 9 Bảng STAFF 33](#_Toc136894986)

[Bảng 2. 10 Bảng STAFF\_GROUP 33](#_Toc136894987)

[Bảng 2. 11 Bảng SALARY 33](#_Toc136894988)

[Bảng 2. 12 Bảng STORE 34](#_Toc136894989)

[Bảng 2. 13 Bảng TIME\_KEEPING 34](#_Toc136894990)

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý nghĩa** |
| FFD | Functional Decomposition Diagram (Sơ đồ phân rã chức năng) |
| CNTT | Công nghệ thông tin |
| CSDL | Cơ sở dữ liệu |
| DFD | Data Flow Diagram (Sơ đồ luồng dữ liệu) |
| ERD | Entiy Relation Diagram (Sơ đồ quan hệ - thực thể) |
| HTTT | Hệ thống thông tin |
| HTTTQL | Hệ thống thông tin quản lý |

PHẦN MỞ ĐẦU

* **Lý do chọn đề tài**

Ngày nay, công nghệ thông tin đang ngày càng phát triển và đóng vai trò quan trọng trong nhiều lĩnh vực của đời sống xã hội. Các ngành nghề, lĩnh vực khác nhau không thể hoạt động dựa vào phương thức quản lý truyền thống như trước mà không có sự can thiệp của CNTT. Những ứng dụng của CNTT nhằm nâng cao hiệu quả công việc trong công tác quản lý, điều hành, sản xuất của từng lĩnh vực, dịch vụ cụ thể. Bởi nó tạo ra hiệu quả công việc cao và giúp tổ chức đạt được những thành tựu đáng kể. Chính vì thế việc thực hiện tin học hoá, số hoá nghiệp vụ quản lý là một nhiệm vụ tất yếu

Có thế nói việc sử dụng các phần mềm, các hệ thống nhắm quản lý chuỗi cửa hàng nhằm tăng hiệu suất quản lý là điều tất yếu. Các thương hiệu muốn hoạt động tốt cần xây dựng một mô hình quản lý chuỗi cửa hàng hoàn chỉnh. Tuy nhiên có nhiều phầm mềm quản lý đã được ứng dụng với nhiều loại ngôn ngữ lập trình, công nghệ khác như: .NET, Ruby on Rails, hay PHP,..

Nhận thấy được vấn đề này, em đã lựa chọn đề tài “Nghiên cứu Spring Boot và Spring Sercurity xây dựng ứng dụng quản lý chuỗi cửa hàng Tous Les Temp” làm đề tài đồ án tốt nghiệp, cũng mong muốn được áp dụng tại đơn vị khảo sát nếu có thể.

* **Nhiệm vụ của đồ án**

Đồ án tập trung vào tìm hiểu và phát triển ứng dụng web service hoàn chỉnh bằng Spring Framework kết hợp sử dụng Angular. Java Spring được sử dụng để phát triển các API và lưu trữ dữ liệu, trong khi Angular có thể được sử dụng để phát triển giao diện người dùng và tương tác với người dùng.

Các nhiệm vụ chính của đồ án là:

* Khảo sát hệ thống đang quản lý chuỗi cửa hàng Tous Les Temp Nha trang.
* Phân tích thiết kế hệ thống.
* Xây dựng cơ sở dữ liệu.
* Xây dựng phần mềm minh họa từ kết quả phân tích và thiết kế hệ thống
* **Bố cục của đồ án**

***Chương 1***: Giới thiệu tổng quan về bài toán và cơ sở lý thuyết

***Chương 2***: Phân tích thiết kế hệ thống cho bài toán quản lý chuỗi cửa hàng.

***Chương 3***: Lập trình ứng dụng

Kết quả nghiên cứu và hướng phát triển.

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN BÀI TOÁN VÀ   
CƠ SỞ LÝ THUYẾT

1.1 Giới thiệu tổng quan về chuỗi cửa hàng tOus Les temps

1.1.1 Một số nét về chuỗi cửa hàng Tous Les Temp

Xuất hiện từ năm 2013, Tous Les Temps đã chinh phục nhiều bạn trẻ từ những cửa hàng đầu tiên để rồi sau đó dần mở rộng thành một chuỗi đồ uống với chuỗi cửa hàng tại Khánh Hòa, Ninh Thuận, Bình Thuận, Thành phố Hồ Chí Minh, Quy Nhơn và trở thành nơi tụ tập, gặp gỡ quen thuộc của giới trẻ nơi đây. Điều nổi bật tại Tous Les Temps là không gian đẹp, thoáng đãng, mỗi góc đều được decor xinh xắn để mỗi lần ghé quán sau những cuộc trò chuyện vui vẻ bên bạn bè chỉ cần chọn một góc bất kỳ để "lên hình", đảm bảo sẽ có ngay vài tấm ảnh nghìn "like"

1.1.2 Tổng quan hiện trạng tại chuỗi cửa hàng Tous Les Temps

Hiện tại, chuỗi cửa hàng hoạt động trong việc bán hàng trực tiếp cho khách hàng tại từng cơ sở, hình thức bán lẻ, chưa thực hiện bán online thông qua các app, chuỗi cửa hàng chú trọng đến chất lượng sản phẩm cùng với chất lượng dịch vụ cho từng cửa hàng, tuy nhiên chưa áp dụng phương pháp quản lý đồng bộ cho chuỗi cửa hàng chi tiết ở các khâu như: quản lý sản phẩm, nhân viên, địa chỉ các chi nhánh, chưa kế thừa dữ liệu trên toàn hệ thống. Để khắc phục việc này, ta phải xây dựng một khung quản lý chuỗi cửa hàng có sự thống nhất, đồng bộ dữ liệu về thành một phần mềm chung. Từ đó, giúp khâu thiết lập menu, giá bán, thông tin nhân sự, mức lương của nhân viên, thống kê số liệu lương nhân viên nhanh chóng

1.1.3 Giải pháp đề xuất cho bài toán quản lý chuỗi cửa hàng

Với thực trạng của Tous Les Temp như hiện nay, để tăng cường hiệu quả trong công tác điểu hành, có thể đưa ra một số giải pháp như sau:

Phát triển một trang web chuyên dụng để quản lý hiệu quả, có khả năng trợ giúp người quản lý trong việc kiểm soát, quản lý danh sách cửa hàng, sản phẩm, danh mục sản phẩm, nhân viên cửa hàng, chấm công, tích hợp các module trên vào mộ thệ thống sẽ giúp cho người quản trị viên dễ dàng trong nghiệp của mình, đạt được mục tiêu cuối cùng đó là mang đến phương án thực sự tối ưu cho chuỗi cửa hàng Tous Les Temp.

1.2 Cơ sở lý thuyết

1.2.1 Một số khái niệm cơ bản

* ***Khái niệm về hệ thống thông tin***

HTTTQL là sự tích hợp có tổ chức của các công nghệ phần cứng và phần mềm, dữ liệu, quy trình và các yếu tố về con người, có nhiệm vụ tìm kiếm, lưu trữ, xử lý và điều phối thông tin. Là một hệ thống tập trung vào việc quản lý CNTT để cung cấp hiệu quả và cung cấp các phương án hiệu quả.

Ngày nay, mảng khoa học nói chung và CNTT nói riêng đang vô cùng phát triển, các doanh nghiệp dần dần đã nhận ra được tầm quan trọng của CNTT và áp dụng nó vào các hoạt động kinh doanh của mình. Vì lí do này, chúng ta phải đẩy mạnh nghiên cứu các HTTTQL có ứng dụng CNTT để thực hiện các hoạt động điều phối, quản lý của tổ chức, doanh nghiệp kinh doanh

* ***Các thành phần của hệ thống thông tin***
* Dữ liệu: Là các thông tin chưa được xử lý (thông tin thô), mới được tìm kiếm và được lưu giữ nói lên tình hình của hệ thống. Các dữ liệu liên về hệ thống: dữ liệu về chứng từ, nhân sự, nguồn lực…
* Các xử lý: Thực hiện chuyển đổi dữ liệu, là giai đoạn chuyển đổi thông tin nhầm cung cấp nguồn thông tin cần thiết.
* Con người: Đây là những người vận hành, sử dụng HTTT để ghi lại các giao dịch hàng ngày. Người sử dụng ở đây thường sẽ là các chuyên gia có trình độ chuyên môn và giữ vai trò quan trọng trong việc triển khai và xây dụng hệ thống.
* Công cụ xử lý: Bao gồm thiết bị như máy tính, máy in, trang thiết bị kết nối mạng. có khả năng tính toán với dữ liệu, đẩy nhanh tốc độ chuyển đổi dữ liệu thành nguồn thông tin.
* Truyền thông: có vai trò quan trọng trong việc kết nối và tương tác giữa các thiết bị và hệ thống trong môi trường mạng. [1]

1.2.2 Quy trình phát triển hệ thống thông tin

* ***Khảo sát hệ thống***

Khảo sát là chu trình khởi đầu trong quy trình phát triển một HTTT. Nhằm nắm bắt những vấn đề hệ thống đang gặp phải để thực hiện giải pháp để khắc phục vấn đề đó. Sau đó sẽ xây dựng kế hoạch cụ thể triển khai phát triển HTTT đã được xác định.

* ***Phân tích hệ thống***

Phân tích hệ thống xử vấn đề bằng cách phân rã các thành chức năng của ứng dụng, hệ thống. Ở bước này, chú trọng tìm hiểu cách thức hoạt động của mỗi thành phần chức năng và sự tác động giữa một thành phần đối với các thành phần chức năng khác.

* ***Thiết kế***

Thiết kế hệ thống căn cứ vào thông tin có được từ giai đoạn khảo sát và phân tích. Dựa trên cơ sở các dữ liệu đã phân tích và đặc tả sẽ được thiết kế dưới dạng mô hình quan niệm. Ngoài ra còn có các bản thiết kế giao diện từng màn hình của chương trình, thiết kế CSDL, truy vấn, hàm, các thông báo, cảnh báo lỗi…

* ***Thực hiện***

Là công đoạn phát triển hệ thống theo những thiết kế đã có được từ những giai đoạn trước. Tiến hành cài đặt, xây dựng cơ sở dữ liệu, các module cho từng chức năng trong ứng dụng

* ***Kiểm thử phần mềm***
* Kiểm thử các chức năng của chương trình
* Triên khai khắc phục các lỗi (nếu có).
* ***Triển khai, bảo trì ứng dụng***
* Thiết lặp các cài đặt phần cứng làm nền tảng cho ứng dụng.
* Triển khai cài đặt phần mềm.
* Chuyển giao hệ thống cho khách hàng, hướng dẫn sử dụng.
* Bảo hành, nâng cấp hệ thống [1]

1.2.3 Mô hình hóa hệ thống thông tin

* ***Sơ đồ luồng dữ liệu (Data Flow Diagram – DFD)***

Sơ đồ luồng dữ liệu (DFD) là một sơ đồ trực quan hóa được sử dụng để minh họa cách dữ liệu được hệ thống thực thi và xử lý. Làm rõ sự di chuyển các luồng thông tin cũng như trình tự các bước để thực hiện một chức năng cụ thể trong ứng dụng. Sơ đồ DFD thể hiện các dòng thông tin, dữ liệu di chuyển và được lưu trữ ra sao.

* ***Sơ đồ quan hệ thực thể (Entiy Relation Diagram – ERD)***

Sơ đồ quan hệ - thực thể là một sơ đồ thể hiện mối quan hệ giữa các tập thực thể được xây dựng trong CSDL. Bên cạnh đó, giải thích cấu trúc logic của CSDL, được xây dụng dựa trên: thuộc tính, thực thể và mối quan hệ giữa chúng. Các bước liên quan đến việc tạo ERD bao gồm:

* Xây dựng các thực thể.
* Xây dựng tất cả các mối quan hệ giữa từng thực thể.
* Phân tích các mối quan hệ.
* Thiết kế sơ đồ ERD mức logic và vật lý [1]

1.3 Phương pháp nghiên cứu

1.3.1 Phương pháp tìm kiếm, khảo sát thông tin

* ***Khảo sát tài liệu về hệ thống***

Khảo sát trên trên nhiều phương diện của tổ chức như lịch sử, hoạt động kinh doanh, lĩnh vực phát triển, hiện trạng cũng như xu hướng của hệ thống tổ chức:

* Xem xét những tài liệu nghiệp vụ đang có sẵn.
* Có thể đưa ra một xem xét lịch sử và chính thức cho các yêu cầu của mô hình.
* ***Quan sát, theo dõi hệ thống***

Là cách lấy thông tin bằng thị giác, xúc giác, .. có liên hệ trực tiếp tới đối tượng khảo sát

* Các thông tin cho nhà phân tích không thu thập được từ các phương pháp khác.
* Quan sát mang lại hiệu quả tốt hơn
* Đáp ứng cho những nghiên cứu thăm dò, dự định.
* ***Phỏng vấn và lắng nghe***

Phỏng vấn và lắng nghe là một trong hai cách lấy thông tin hay dùng cho các hệ thống thông tin hiện nay. Dựa vào việc lắng nghe nhu cầu thực tế của khách hàng

* Tập hợp các sự kiện, ý kiến và suy đoán.
* Chú ý vào ngôn ngữ cơ thể và những biể cảm của người được phỏng vấn.
* *Phỏng vấn cá nhân*: giúp các nhà tuyển dụng hoặc phân tích viên đánh giá được năng lực ứng viên được hỏi chính xác nhưng đồng thời cũng gặp phải một số điểm trừ như sau: những cuộc phỏng vấn mới có thể tiết lộ những câu hỏi mới cần phỏng vấn thêm với những người đã thực hiện trước đó
* *Phỏng vấn nhóm*: Hình thức này sẽ đem lại một số ưu điểm gồm: hiệu quả hơn về thời gian, có thể nghe được sự đồng thuận hoặc mâu thuẫn cùng một lúc, cơ hội cho sự hợp lực. Bên cạnh đó hình thức này còn tồn tại sự khó khăn trong việc lập lịch vì mỗi người có một thời gian nhất định. [2]

1.3.2 Các phương pháp phân tích, thiết kế hệ thống

Một hệ thống được phát triển sẽ dựa trên các phương pháp phân tích và thiết kế khác nhau. Dưới đây là hai hướng để phân tích hay dùng:

|  |  |  |
| --- | --- | --- |
| **Những đặc điểm** | **Hướng chức năng** | **Hướng đối tượng** |
| Phương pháp luận | SDLC – chu trình phát triển của hệ thống | Lặp lại/ tăng trưởng |
| Tập trung | Các tiến trình | Các đối tượng |
| Rủi ro | Cao | Thấp |
| Sử dụng lại | Thấp | Cao |
| Tính hoàn thiện | Hoàn thiện và mở rộng | Đang phát triển |
| Phù hợp với | Những dự án được định nghĩa rõ ràng với những yêu cầu người dùng ổn định | Những dự án lớn nhiều rủi ro với những thay đổi về các nhu cầu của khách hàng |

1.4 Môi trường lập trình

1.4.1 IntelliJ IDEA

IntelliJ IDEA là một IDE (Integrated Development Environment) cho ngôn ngữ Java được phát triển và nâng cấp bởi JetBrains. Cung cấp một môi trường phát triển tích hợp đầy đủ cho các lập trình viên Java, bao gồm tính năng đánh dấu cú pháp,  
auto-complete, refactoring, kiểm tra lỗi cú pháp và xử lý lỗi, kiểm tra đơn vị, quản lý phiên bản và các tính năng khác. Nó giúp bạn làm việc hiệu quả với một bộ các tính năng nâng cao như hỗ trợ viết mã thông minh, tái cấu trúc đáng tin cậy

* Tính năng của IntelliJ IDEA

IntelliJ IDEA hỗ trợ các kỹ thuật lập trình hiện đại như Spring, Hibernate, Grails, Struts, Play Framework, Kotlin và Groovy. Nó cũng cung cấp tính năng đáng chú ý như hỗ trợ trực quan cho GUI designer, kiểm soát tính bảo mật, tối ưu hóa mã, tạo ra tài liệu tự động và hỗ trợ toàn diện các khung cấu hình, quản lý dự án như Maven và Gradle.

* Built-in tools và integrations

IntelliJ IDEA có rất nhiều built-in tools và tích hợp với nhiều công cụ khác nhau, tạo điểu kiện xây dựng, phát triện ứng dụng nhanh chóng, giúp bạn làm việc trong một môi trường quen thuộc

* JVM (Java Virtual Machine) framworks

IntelliJ IDEA Ultimate cung cấp hỗ trợ tốt nhất cho các framworks và công nghệ hàng đầu chú trọng đến sự phát triển, hình thành của các ứng dụng và microservices, dành riêng cho Spring, Spring Boot, Jakarta EE, JPA, Reactor và các framworks khác. [3]

1.4.2 Microsoft Visual Studio Code

Visual Studio Code là trình chỉnh sửa mã nguồn miễn phí, nhanh và mạnh mẽ. Cung cấp một môi trường lập trình mạnh mẽ với hệ sinh thái các hỗ trợ dồi dào. Nhiều tiện ích được cập nhật thường xuyên khi cần thiết. Mức độ phong phú của hỗ trợ khác nhau giữa các ngôn ngữ lập trình khác nhau và phần mở rộng của chúng [5]

1.4.3 Postman

Postman là một ứng dụng để phát triển và sử dụng API. Nó giúp cho lập trình viên có thể nhanh chóng thực hiện các cú pháp với API đã được xây dựng nhằm kiểm tra kết qua các chức năng của API

1.4.4 PowerDesigner

PowerDesigner là ứng dụng được dùng để thiết kế các sơ đồ hỗ trợ cho việc phân tích hệ thống, tạo một số sơ đồ như ERD, DFD, … Từ đó có cái nhìn tổng quan cho hệ thống được phân tích với nhưng sơ đồ có được đảm bảo cho việc thực hiện đúng đủ các chức năng đã được đề ra trong giai đoạn xây dựng chương trình

1.5 Công nghệ sử dụng

1.5.1 Spring Boot Framework

1.5.1.1 Giới thiệu Spring Boot

Spring Boot giúp bạn tạo ra các ứng dụng độc lập, cung cấp cơ chế hoạt động tương tích với môi trường phát triển và dựa trên Spring Framework.

Stand-alone: Spring Boot cho phép bạn tạo ra các ứng dụng độc lập, có thể thực thi mà không cần sự phụ thuộc vào một máy chủ ứng dụng ngoài. Điều này giúp đơn giản hóa triển khai và triển khai ứng dụng Spring Boot trên nhiều môi trường khác nhau một cách dễ dàng.

Production-grade: Cung cấp các tính năng và cấu hình tiêu chuẩn để xử lý các khía cạnh quan trọng của ứng dụng, bao gồm bảo mật, hiệu suất, quản lý lỗi và khả năng mở rộng.

Spring-based: Spring Boot dựa trên Spring Framework, nó cung cấp nhiều tính năng và module để xây dựng ứng dụng Java theo mô hình phát triển ứng dụng diễn dịch (IoC) và phần tử phần mềm (AOP).

* Tại sao chọn Spring
* Spring ở khắp mọi nơi:

Các thư viện linh hoạt của Spring được các nhà phát triển trên toàn thế giới tin cậy. Spring cũng được tích hợp trong lĩnh vực khoa học công nghệ, bao gồm Alibaba, Amazon, Google, Microsoft, v.v.

* Spring có tính linh hoạt:

Về cốt lõi, các tính năng Inversion of Control (IoC) và Dependency Injection (DI) của Spring Framework cung cấp nền tảng cho nhiều tính năng. Cho dù bạn đang xây dựng các vi dịch vụ an toàn, tương tác, dựa trên đám mây cho web hay các luồng dữ liệu truyền trực tuyến phức tạp cho doanh nghiệp, thì Spring đều có các công cụ để trợ giúp.

* Spring mang lại năng suất hơn, nhanh hơn

Spring Boot biến đổi cách tiếp cận các tác vụ lập trình Java, Spring Boot kết hợp các yếu tố cần thiết như ngữ cảnh ứng dụng và máy chủ web nhúng, được định cấu hình tự động để giúp quá trình phát triển microservice trở nên dễ dàng. Để đi nhanh hơn nữa, ta có thể kết hợp Spring Boot với bộ thư viện, máy chủ, mẫu và mẫu hỗ trợ phong phú của Spring Cloud để triển khai toàn bộ kiến trúc dựa trên vi dịch vụ vào đám mây một cách an toàn.

* Spring có tính bảo mật:

Spring có thành tích đã được chứng minh về việc khắc phục các khuyết điểm về bảo mật tối ưu. Các ủy viên của Spring làm việc với các chuyên gia bảo mật để vá và kiểm tra bất kỳ lỗ hổng nào được báo cáo. Sự phụ thuộc của bên thứ ba cũng được giám sát chặt chẽ và các bản cập nhật thường xuyên được phát hành để giúp giữ cho thông tin dự án toàn nhất có thể. [4]

1.5.1.2 Yêu cầu hệ thống

Spring Boot 3.0.0 yêu cầu Java 17 và tương thích với Java 20. Spring Framework 6.0.0, Maven 3.6.0, Gradle 7x trở lên cũng được yêu cầu .

Maven là công cụ build quản lý phụ thuộc (dependency management) dùng trong lập trình Java. Bạn có thể sử dụng Maven để quản lý các phụ thuộc của dự án Spring Boot và xây dựng (build) ứng dụng.

Gradle là một công cụ build linh hoạt và mạnh mẽ, cung cấp khả năng cấu hình linh hoạt hơn so với Maven. Thường được cấu hình trong các ứng dụng Spring. Gradle sử dụng Domain Specific Language để định nghĩa quy trình build và quản lý phụ thuộc.

1.5.1.3 Cài đặt Spring Boot

Spring Boot có thể được sử dụng với điều kiện là Java SDK 8 trở lên. Có thể sử dụng Spring Boot giống như bất kỳ thư viện Java tiêu chuẩn nào. Để làm như vậy, hãy bao gồm các tệp spring-boot-\*.jar thích hợp trên đường dẫn classpath của bạn. Spring Boot không yêu cầu tích hợp bất kỳ công cụ đặc biệt nào, vì vậy bạn có thể sử dụng bất kỳ IDE hoặc trình soạn thảo văn bản nào. Ngoài ra, không có gì đặc biệt về ứng dụng Spring, vì vậy bạn có thể chạy và gỡ lỗi ứng dụng Spring Boot giống như bất kỳ chương trình Java nào khác. Tiếp theo tiến hành cài đặt Maven bằng dòng lệnh sudo apt-get install maven hoặc Gradle từ https://gradle.org/releases/ [6]

1.5.2 Spring Security Framework

Spring Security là một khung xác thực và kiểm soát truy cập mạnh mẽ và có khả năng tùy biến cao. Spring Security framework chú trọng vào hỗ trợ xác thực và   
phân quyền cho các dự án Java. Giống như tất cả các dự án Spring, sức mạnh của Spring Security nằm ở việc nó có thể được mở rộng dễ dàng như thế nào để đáp ứng các yêu cầu tùy chỉnh

* Đặc điểm của Spring Security Framework
* Hỗ trợ toàn diện và có thể mở rộng cho cả xác thực và phân quyền
* Bảo vệ chống lại các cuộc tấn công như session fixation
* Tích hợp API Servlet, tích hợp tùy chọn với Spring Web MVC
* Spring Security yêu cầu Môi trường thời gian chạy Java 8 trở lên.
* Vì Spring Security nhằm mục đích hoạt động theo cách khép kín, bạn không cần đặt bất kỳ tệp cấu hình đặc biệt nào trong Môi trường chạy thi hành Java của mình. Cụ thể, bạn không cần định cấu hình tệp chính sách dịch vụ xác thực và ủy quyền Java (JAAS) đặc biệt hoặc đặt Spring Security vào các vị trí đường dẫn chung.
* Ở dự án lần này, tôi tích hợp Spring Security 6 and Spring Boot 3 để áp dụng vào hệ thống web quản lý chuỗi cửa hàng Tous Les Temp [6]

1.5.3 Angular

Angular là một framework thiết kế ứng dụng và nền tảng phát triển để tạo các ứng dụng một trang single-page app hiệu quả. Cung cấp một môi trường lập trình với các thư viện có sẵn nhằm đẩy nhanh quá trình cấu hình. Một số gói hay được dùng như: angular cli, angular material, primeNG,..

* Ứng dụng Angular: Các yếu tố cần thiết
* Components
* Templates
* Directives
* Dependency injection

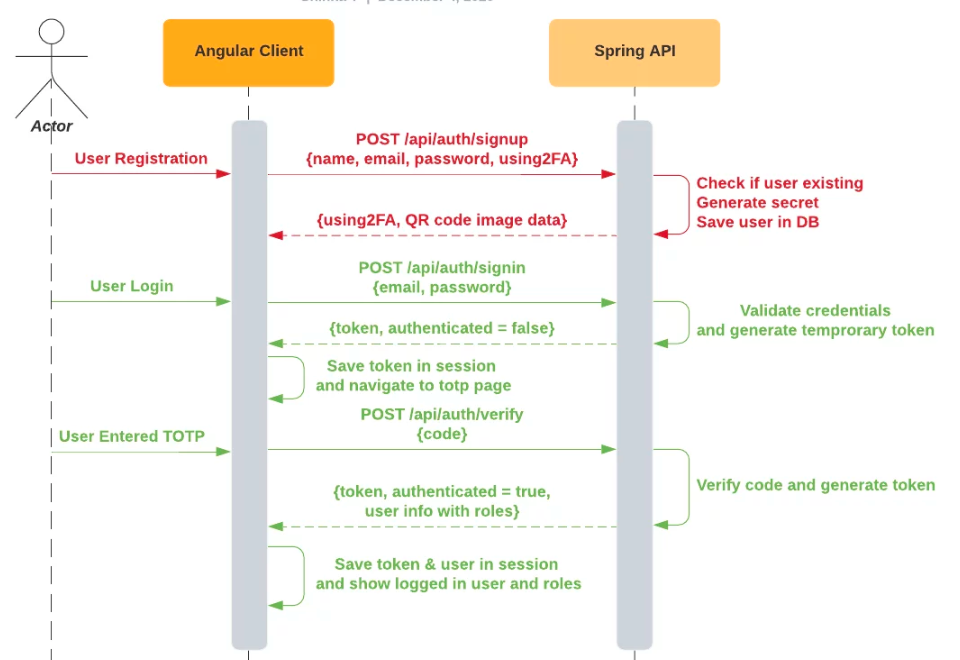
1.5.4 Two Factor Authentication

1.5.4.1 Two Factor Authentication là gì ?

**2FA** Two-factor authentication là phương pháp xác thực hai yếu tố, thực hiện tạo hai lớp bảo mật cho ứng dụng, đảm bảo thông tin của người dùng đăng trang web, hệ thống, thông qua việc yêu cầu nhập mã xác thực OTP sau khi thực hiện đăng nhập với thông tin đăng nhập cơ bản trước khi truy cập đến nguồn tài nguyên của hệ thống

1.5.4.2 Ưu điểm xác thực hai yếu tố 2FA

* Nâng cao khả năng bảo mật của hệ thống
* Trong trường hợp mất tài khoản, thông tin của người dùng được bảo mật
* Khắc phục các thiếu sót của phương pháp đăng nhập thông thường

1.5.4.3 Cách thức hoạt động của 2FA ?

Hình 1. Cơ chế hoạt động của 2FA

Ở dự án này, ta sẽ triển khai xác thực 2FA với Spring Security để thực hiện 3 thao tác:

* Tạo JWT – Khi tên đăng nhập và mật khẩu chính xác, nếu đã kích hoạt dùng 2FA trong quá trình đăng ký, thì nó sẽ tạo một JSON Web Token (JWT), đồng đặt cờ đã xác thực vào mã thông báo để cho biết rằng chưa được xác thực đầy đủ.
* Xác thực mã thông báo mềm (TOTP) – Khởi tạo API POST với ánh xạ “/verify”. Khi chuyển mã thông báo chính xác, nó sẽ tạo một JWT với thời gian hết hạn là 10 ngày và đính kèm trả lại JWT token này trong respone cùng dữ liệu tài khoản
* Xác thực mã JWT – Nếu người dùng cố truy cập API, nó sẽ chỉ cho phép truy cập nếu yêu cầu có token hợp lệ. Nếu cờ được xác thực trong mã thông báo là sai, thì chỉ vai trò PRE\_VERIFICATION\_USER mới được cấp cho người dùng. Với vai trò này, chỉ có khả năng truy cập endpoint “/verify”.

Thêm Spring Security vào ứng dụng:

* Thêm dependency Spring Security vào file pom.xml hoặc build.gradle của dự án.
* Cấu hình Spring Security trong file cấu hình (SecurityConfig.java) của ứng dụng.

Xác thực người dùng bằng một yếu tố đầu tiên (tên đăng nhập và mật khẩu) thông qua việc sử dụng các thức xác thực mặc định của Spring Security.

Triển khai xác thực 2 yếu tố:

* Tạo form đăng ký, người dùng kích hoạt sử dụng 2FA để xác thực
* Sử dụng ứng dụng Authenticator để quét QR code để tạo soft token
* Tạo form đăng nhập để người dùng nhập mã OTP
* Xác thực mã OTP và kiểm tra tính hợp lệ của nó
* Cung cấp quyền truy cập vào ứng dụng nếu xác thực thành công

2FA là phương án để tăng cường bảo mật tài khoản và dữ liệu cá nhân, và nó được sử dụng rộng rãi ở nhiều hệ thống như: ứng dụng di động, ngân hàng, dịch vụ đám mây, và nhiều hệ thống trực tuyến khác.

1.5.5 Google Map API

Google Maps API là tập hợp các công cụ và các API được Google cung cấp để tích hợp và sử dụng dữ liệu và chức năng của Google Maps trong các ứng dụng và trang web của bạn. API này cho phép bạn tương tác với dịch vụ Google Maps, như hiển thị bản đồ, tìm kiếm địa điểm, tạo chỉ đường, tính khoảng cách và thực hiện nhiều chức năng khác.

Google Maps API cung cấp các phiên bản khác nhau dựa trên mục đích sử dụng của bạn, bao gồm:

* Google Maps JavaScript API: Cho phép bạn nhúng và tương tác với bản đồ Google Maps trong ứng dụng web của mình bằng cách sử dụng JavaScript.
* Google Maps Geocoding API: Cung cấp các phương thức để chuyển đổi địa chỉ sang tọa độ địa lý (latitude và longitude) và ngược lại.
* Google Maps Directions API: Cho phép bạn tạo chỉ đường giữa các địa điểm, tính toán thời gian và khoảng cách dựa trên các tùy chọn như phương tiện di chuyển, điểm dừng, và hạn chế đường.
* Google Maps Static API: Cho phép bạn tạo và tải về hình ảnh tĩnh của bản đồ Google Maps, điểm đánh dấu và lớp thông tin.

1.5.6 Swagger

Swagger được tích hợp nhầm xây dựng một tài liệu rõ ràng và dễ đọc cho các API. Nó mô tả các điểm cuối (endpoints) của API, các phương thức hỗ trợ, các tham số của mỗi hàm chức năng, các mô hình dữ liệu, và các thông tin khác về API. Tạo ra tài liệu tự động, tạo giao diện người dùng tương tác với API, và thậm chí tạo mã khách hàng (client code) cho các ngôn ngữ lập trình khác nhau. Từ đây, ta có thể thao tác với các API đã được xây dựng trong môi trường lập trình một cách nhanh chóng, không cần thông qua Postman

* Thêm các dependency sau tiến hành cấu hình Swagger cho dự án



Hình 1. Swagger dependency

* Cấu hình Swagger trong file **SwaggerConfig.java**



Hình 1. Class SwaggerConfig.java

* Swagger sẽ tự động quét và tạo tài liệu API dựa trên các annotation như @RestController, @RequestMapping, và các chú thích khác được sử dụng trong ứng dụng Spring Boot. Bạn có thể tùy chỉnh và mở rộng Swagger để bổ sung thông tin bổ sung về API như mô tả, tham số, ví dụ và kiểu dữ liệu.
* Khởi động ứng dụng và mở trình duyệt và truy cập vào URL http://localhost:8080/your context-path/swagger-ui.html.để kiểm tra tài liêu API

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG CHO BÀI TOÁN QUẢN LÝ CHUỖI CỬA HÀNG tOUS LES TEMP

2.1 Đặc tả hệ thống

Hệ thống quản lý chuỗi cửa hàng Tous Les Temp là hệ thống quản lí một chuỗi những các chi nhánh Tous Les Temp. Việc thực hiện quản lý dựa trên danh sách các cửa hàng Tous Les Temp ở Nha Trang, dễ dàng mở rộng quản lý cho các khu vực khác, tạo nên cái nhìn tổng quan cho quản lý về việc thực hiện quản lý sản phẩm, nhân sự và thông tin chuỗi cửa hàng của hệ thống một cách đồng bộ, nhanh chóng

2.1.1 Quản lý danh sách cửa hàng

* Yêu cầu chức năng
* Quản lí thông tin cửa hàng
* Thục hiện thêm, sửa đổi và xoá thông tin cửa hàng
* Hỗ trợ việc tải và lưu trư ảnh của cửa hàng ở firebase
* Tìm kiếm cửa hàng
* Tìm kiếm thông tin cửa hàng theo nhiều tiêu chí
* Hệ thống thực hiện sử dụng Google Map API để quản lí địa chỉ của từng cửa hàng của hệ thống, khi chọn cửa hàng nào thì hiển thị map cho địa chỉ của cửa hàng, nhờ đó việc tìm kiếm, theo dõi cửa hàng sinh động hơn
* Xem danh sách cửa hàng
* Cho phép hiển thị tấc cả cửa hàng có sẵn, cung cấp thông tin địa chỉ, số điện thoại,…
* Xem thông tin chi tiết số lượng sản phẩm có trong từng chi nhánh
* Thực hiện phân trang khi số lượng cửa hàng lớn có nhiều bản ghi
* Yêu cầu phi chức năng
* Bảo mật:
* Thực hiện các biện pháp bảo mật kiểm soát chỉ những người dùng có quyền truy cập mới có thể thao tác trên danh sách cửa hàng.
* Hệ thống nên hỗ trợ cơ chế xác thực an toàn như Two-Factor Authentication (2FA) để bảo vệ tài khoản quản trị.
* Hiệu suất:
* Hệ thống cần đảm bảo khả năng xử lý và hiển thị thông tin cửa hàng một cách nhanh chóng và hiệu quả.
* Dễ sử dụng:
* Giao diện người dễ sử dụng và thân thiện với người dùng

2.1.2 Quản lý sản phẩm

* Yêu cầu chức năng
* Quản lý sản phẩm
* Thêm, sửa đổi và xoá thông tin sản phẩm
* Thực hiện lưu trữ hình ảnh lên firebase để tối ưu bộ nhớ cho database
* Tìm kiếm, lọc sản phẩm
* Tìm kiếm sản phẩm theo nhiều tiêu chí: tên sản phẩm, loại sản phẩm, số lượng sản phẩm, đơn giá,…
* Hiển thị thông tin lọc chính xác, có thực hiện phân trang nếu kết quả tìm kiếm nhiều bản ghi
* Xem danh sách sản phẩm của từng cửa hàng
* Hiển thị danh sách sản phẩm có sẵn của từng cửa cửa hàng nhằm cập nhật thông tin tiện lợi
* Xem thông tin chi tiết số lượng sản phẩm có trong từng chi nhánh
* Yêu cầu phi chức năng
* Bảo mật:
* Thực hiện các biện pháp bảo mật kiểm soát chỉ những người dùng có quyền truy cập mới có thể thao tác trên danh sách sản phẩm.
* Hệ thống nên hỗ trợ cơ chế xác thực an toàn như Two-Factor Authentication (2FA) để bảo vệ tài khoản quản trị.
* Hiệu suất:
* Đảm bảo khả năng xử lý các API phía server và hiển thị thông tin sản phẩm một cách nhanh chóng và hiệu quả.
* Dễ sử dụng:
* Giao diện người dùng dễ sử dụng và thân thiện với người dùng,
* Dễ dàng tìm kiếm theo nhiều tiêu chí

2.1.3 Quản lý nhân viên

* Yêu cầu chức năng
* Quản lý thông tin nhân viên
* Cho phép người dùng thêm, sửa đổi và xoá thông tin nhân viên
* Thực hiện lưu trữ hình ảnh lên firebase để tối ưu bộ nhớ cho database
* Tìm kiếm nhân viên
* Thực hiện tìm kiếm nhân viên theo tên
* Hiển thị thông tin lọc chính xác, có thực hiện phân trang nếu kết quả tìm kiếm nhiều bản ghi
* Xem danh sách nhân viên
* Hiển thị tấc cả nhân viên của chuỗi cửa hàng, cập nhật thông tin tiện lợi
* Người dùng có thể xem chi tiết bảng chấm công của từng nhân viên, từ đó dễ dàng cho việc kiếm soát chấm công
* Yêu cầu phi chức năng
* Bảo mật:
* Thực hiện biện pháp bảo mật để đảm bảo rằng chỉ những người dùng có quyền truy cập mới tác trên danh sách nhân viên.
* Hệ thống nên hỗ trợ cơ chế xác thực an toàn như Two-Factor Authentication (2FA) để bảo vệ tài khoản quản trị.
* Hiệu suất:
* Cung cấp khả năng xử lý các API phía server và hiển thị thông tin nhân viên một cách nhanh chóng và hiệu quả.
* Dễ sử dụng:
* Giao diện dễ sử dụng, giúp người dùng dễ dàng thao tác và tìm kiếm thông tin cần thiết

2.1.4 Tra cứu thông tin

* Yêu cầu chức năng:
* Tìm kiếm theo từ khóa:
* Có thể nhập từ khóa để tìm kiếm thông tin.
* Tìm kiếm dựa trên các trường thông tin như tên, mô tả, danh mục, v.v.
* Tìm kiếm theo tiêu chí lọc:
* Có nhiều tiêu chí lọc cho người dùng, ví dụ: danh mục, giá, loại sản phẩm, cửa hàng, kích thước sản phẩm
* Thực hiện một hoặc nhiều tiêu chí lọc để hạn chế kết quả tìm kiếm kết quả theo mong muốn của mình.
* Xử lý kết quả tìm kiếm:
* Nên xử lý phù hợp với từ khóa và tiêu chí lọc đã nhập.
* Kết quả tìm kiếm nên bao gồm thông tin cơ bản về các mục được tìm thấy, ví dụ: tên, mô tả, hình ảnh, v.v.
* Thực hiện phân trang cho kết quả tìm kiếm
* Sắp xếp kết quả được tìm kiếm:
* Hệ thống nên cho phép sắp xếp kết quả tìm kiếm theo một tiêu chí
* Có thể chọn thứ tự tăng dần hoặc giảm dần cho tiêu chí sắp xếp.
* Dùng RSQL cho việc tìm kiếm và lọc dữ liệu. RSQL sử dụng cú pháp đơn giản tạo ra các biểu thức truy vấn phức tạp mà không cần phải viết mã SQL tường minh.

Yêu cầu phi chức năng:

* Hiệu suất:
* Cung cấp khả năng xử lý các API phía backend và hiển thị kết quả tìm kiếm một cách nhanh chóng và hiệu quả, chính xác

2.1.5 Chấm công nhân viên

* Yêu cầu chức năng:
* Tạo bảng chấm công:
* Cho phép người quản lý tạo, cập nhật bảng chấm công nhân viên, từ đó thực hiện lưu trữ dữ liệu và thực hiện tính lương nhân viên
* Lưu trữ ngày thực hiện chấm công
* Thực hiện tính lương:
* Hệ thống sẽ tính lương nhân viên dựa trên các bảng chấm công mà nhân viên có, từ đó tính lương theo tháng hoặc tổng lương
* Tiền lương sẽ được tính theo lương cơ bản của từng vị trí trong cửa hàng và theo công thức cụ thể
* Báo cáo kết quả chấm công:
* Hệ thống nên lưu trữ kết quả quá trình chấm công, từ đó sử dụng thư viện tạo biểu đồ trong Angular để thông kê

Yêu cầu phi chức năng:

* Dễ sử dụng:
* Biểu đồ khi được kết xuất phải thể hiện đầy đủ, chính xác, thông tin của lương nhân viên
* Màu sắc tương quan giữa các cột sinh động, rõ ràng

2.1.6 Báo cáo thống kê

Yêu cầu chức năng:

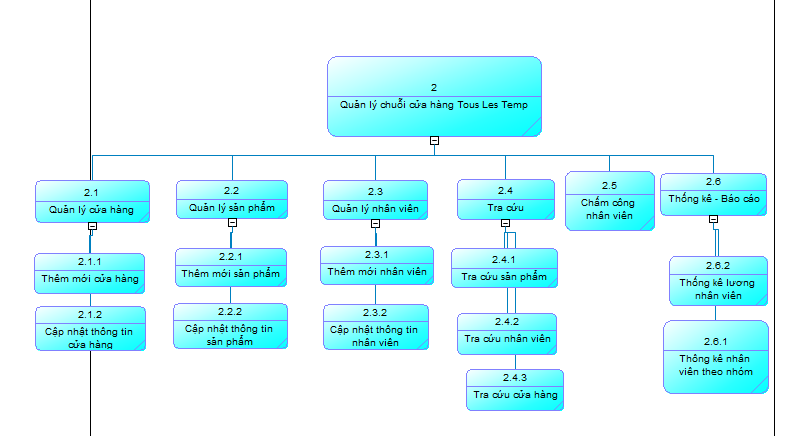
* Thực hiện tạo biểu đồ thống kê lương nhân viên
* Thực hiện kết xuất dữ liệu chính xác
* Biểu đồ thống kê sẽ hiển thị lương từng nhân viên theo tháng, giúp người quán lý có cái nhìn trực quan về chấm công
* Việc thực hiện thống kê bằng biểu đồ là điểm nổi bật trong dự án này, kết xuất được dữ liệu từ phía backend, truyền lên cho frontend xử lí gắn dữ liệu cho biều đồ phải được thực hiện chỉnh chu, tránh nhầm lẫn gây lỗi ngoài mong muốn
* Thực hiện tạo biểu đồ thống kê số lượng nhân sự theo từng vai trò
* Phân tích số lượng nhân sự để xây dựng biểu đồ cho từng nhóm nhân sự

Yêu cầu phi chức năng:

* Dễ sử dụng:
* Giao diện biểu đồ khi được kết xuất phải thể hiện đầy đủ, chính xác, thông tin của lương nhân viên
* Màu sắc tương quan giữa các cột sinh động, rõ ràng

2.2 Sơ đồ mô tả hệ thống

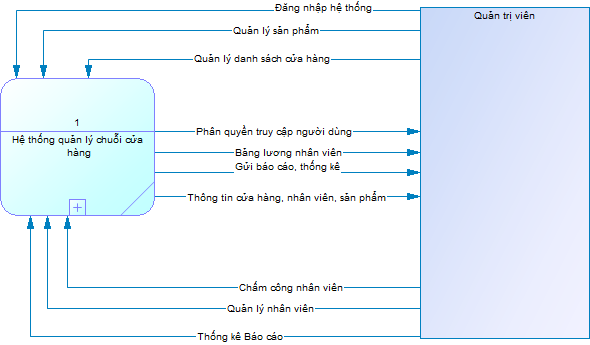
2.2.1 Sơ đồ phân rã chức năng



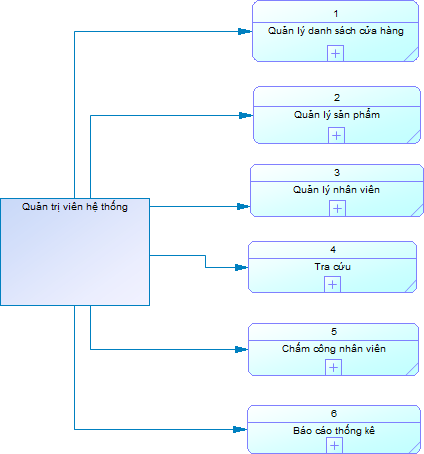
HInh 2. Sơ đồ phân rã chức năng - FDD

* ***Các chức năng của sơ đồ FFD***
* Quản lí cửa hàng:
* Quản lý thông tin về tấc cả cửa hàng, thêm mới, cập nhật và xóa cửa hàng.
* Chức năng cũng cung cấp khả năng xem danh sách các cửa hàng và xem chi tiết từng cửa hàng.
* Quản lí nhân viên:
* Quản lí thông tin về nhân viên, thêm mới, cập nhật và xóa nhân viên.
* Chức năng cũng cung cấp khả năng xem thông tin tấc cả nhân viên và xem chi tiết từng nhân viên.
* Quản lí sản phẩm:
* Quản lý thông tin về sản phẩm, thêm mới, cập nhật và xóa sản phẩm.
* Chức năng cũng cung cấp khả năng xem danh sách sản phẩm và xem chi tiết từng sản phẩm.
* Chấm công:
* Chức năng này cho phép chấm công nhân sự
* Ghi lại số ca làm, vị trí làm việc từ đó tạo bảng chấm công và tính tiền lương, lưu trữ thông tin lương cung cấp cho việc tạo biểu đố
* Xem lịch sử chấm công của mỗi nhân viên.
* Tra cứu:
* Tra cứu, tìm kiếm cửa hàng, nhân viên hoặc sản phẩm.
* Triển khai trên nhiều tiêu chí như tên, địa chỉ, tên nhân viên, tên sản phẩm
* Kết quả tra cứu sẽ hiển thị danh sách phù hợp với tiêu chí tìm kiếm.
* Thống kê:
* Xem các báo cáo và biểu đồ thống kê liên quan đến lương của nhân viên, số lượng nhân viên của cửa hàng.

2.2.2 Sơ đồ phân rã mức ngữ cảnh

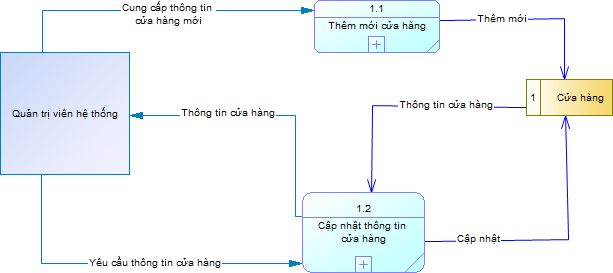


HInh 2. Sơ đồ mức ngữ cảnh quản lý chuỗi cửa hàng

2.2.3 Sơ đồ phân rã mức 0

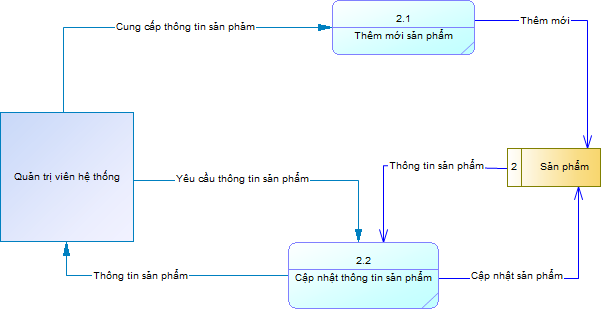
HInh 2. Sơ đồ phân rã DFD mức 0

2.2.4 Sơ đồ phân rã mức 1 Quản lý danh sách cửa hàng



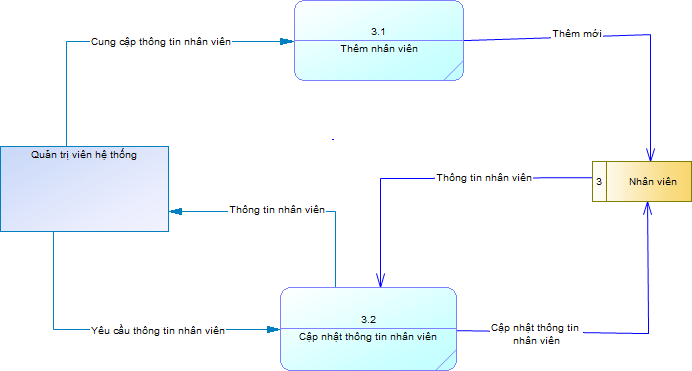
HInh 2. Sơ đồ phân rã DFD mức 1 Quản lý danh sách cửa hàng

2.2.5 Sơ đồ phân rã mức 1 Quản lý danh sách sản phẩm



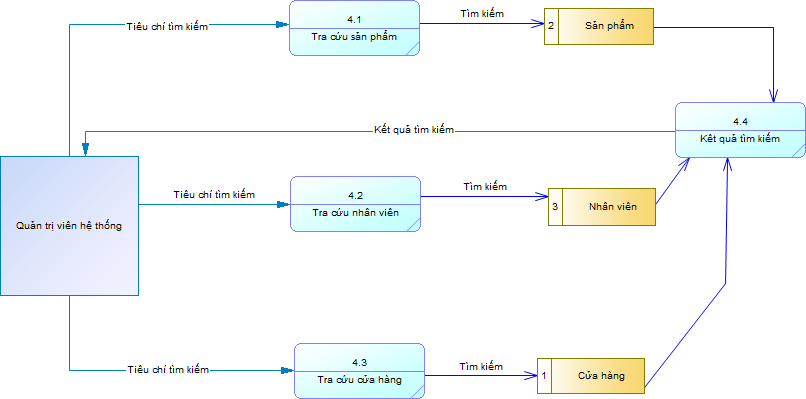
HInh 2. Sơ đồ phân rã DFD mức 1 Quản lý sản phẩm

2.2.6 Sơ đồ phân rã mức 1 Quản lý nhân viên



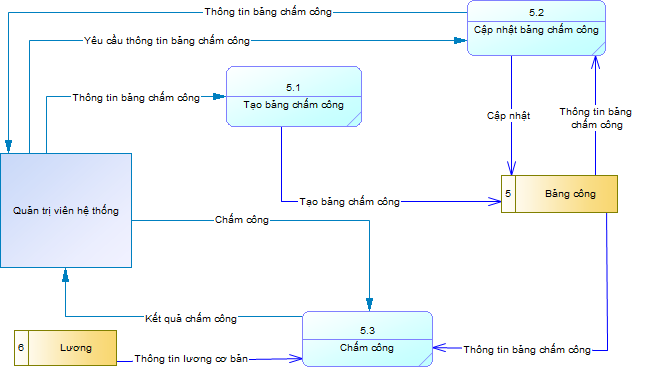
HInh 2. Sơ đồ phân rã DFD mức 1 Quản lý nhân viên

2.2.7 Sơ đồ phân rã mức 1 Tra cứu



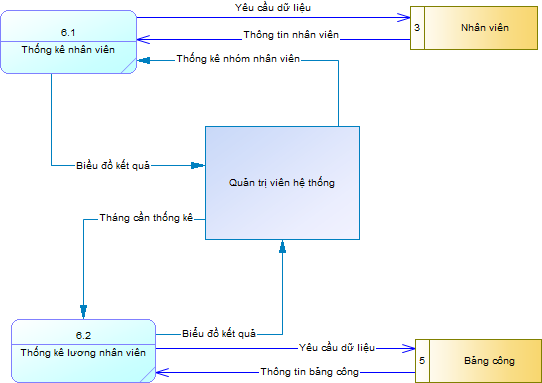
HInh 2. Sơ đồ phân rã DFD mức 1 Tra cứu

**2.2.8 Sơ đồ phân rã mức 1 Chấm công nhân viên**



HInh 2. Sơ đồ phân rã DFD mức 1 Chấm công nhân viên

2.2.9 Sơ đồ phân rã mức 1 Báo cáo thống kê



HInh 2. Sơ đồ phân rã DFD mức 1 Báo cáo thống kê

2.3 Đánh giá hệ thống dữ liệu

2.3.1 Phân tích hệ thống quản lý chuỗi cửa hàng Tous Les Temp

Trong khung quản lý này, phân quyền truy cập ứng dụng là quá trình xác định và quản lý quyền truy nhập vào hệ thống căn cứ từ vai trò và quyền hạn của đối tượng. Đảm bảo toàn vẹn và an toàn dữ liệu.

Hệ thống quản lý chuỗi cửa hàng Tous Les Temp là mô hình được thiết kế để quản lý, xây dựng chuỗi cửa hàng trà sữa. Hệ thống này giúp cho việc quản lý hoạt điều phối, đảm bảo chất lượng và tăng cường hiệu quả vận hành của chuỗi cửa hàng. Kiểm soát được quá trình thêm mới chi nhánh cũng như sản phẩm và nhân viên từng chi nhánh

Hệ thống cung cấp các xử lí kiếm soát thông tin từng cửa hàng về địa điểm, sản phẩm, nhân viên. Bao gồm các tính năng như quản lý danh sách chuỗi cửa hàng ở từng khu vực.

Quản lý sản phẩm: Hệ thống giúp theo dõi và quản lý thông tin về sản phẩm trà sữa, bao gồm danh sách sản phẩm, mô tả, giá cả, loại sản phẩm và số lượng hiện có. Hiện tại, hệ thống chưa thực hiện chức năng có thể hỗ trợ quản lý nguồn cung cấp và đặt hàng tự động khi số lượng tồn đạt ngưỡng quy định.

Quản lý nhân viên: Hệ thống cho phép quản lý thông tin về nhân viên, vị trí công việc, bao gồm lịch làm việc, chấm công, tính lương và thông tin nhân viên.

Quản lý lịch làm việc: Hệ thống cung cấp tính năng quản lý lịch làm việc cho nhân viên. Quản lý có thể tạo và chỉnh sửa lịch làm việc, bao gồm thời gian làm, số ca làm của từng nhân viên từ đó hệ thống có thể tính toán lương dựa trên thông tin chấm công. Nó tính toán số giờ làm và áp dụng các quy tắc tính lương

Báo cáo chấm công: Thực hiện báo cáo chấm công. Báo cáo này một biểu đồ bao gồm thông tin về số giờ làm việc của từng nhân viên mức lương nhân viên nhận được theo tháng. Báo cáo giúp quản lý kiểm tra hiệu suất làm việc của nhân viên và đưa ra quyết định về lương, chính sách làm việc và quản lý nhân sự.

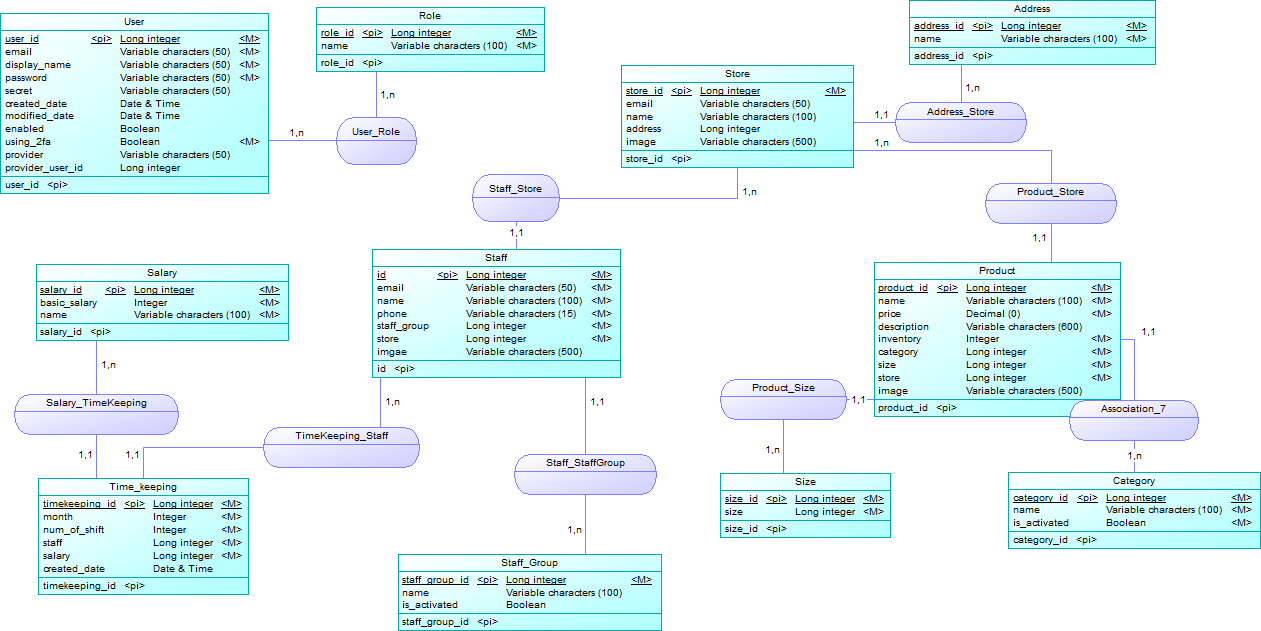
2.3.2 Các thực thể liên kết

Bảng 2. Các thực thể, định danh và thuộc tính mô tả

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thực thể** | **Định danh** | **Thuộc tính mô tả** |
| 1 | User | user\_id | created\_date  modified\_date  display\_name  password  email  enable  provider  provider\_user\_id  secret  using2fa |
| 2 | Role | role\_id | name |
| 3 | User\_Role | user\_id  role\_id |  |
| 4 | Address | address\_id | name |
| 5 | Store | store\_id | name  email  phone  address  address\_detail  image |
| 6 | Product | product\_id | name  description  inventory  price  category  size  store  image |
| 7 | Category | category\_id | is\_activated  name |
| 8 | size | size\_id | size |
| 9 | Staff | staff\_id | name  email  phone  staff\_group  store  image |
| 10 | Staff\_group | staff\_group\_id | is\_activated  name |
| 11 | Time\_keeping | time\_keeping\_id | month  num\_of\_shift  staff  salary  created\_date |
| 12 | salary | salary\_id | name  basic\_salary |

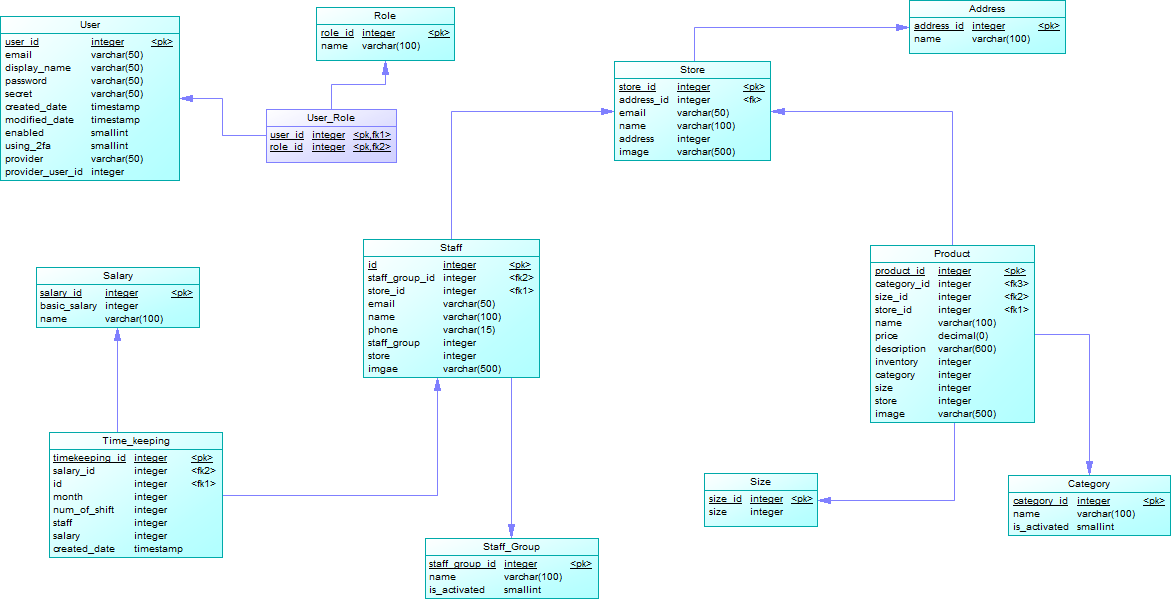
2.3.3 Biến đổi sơ đồ thực thể liên kết sang sơ đồ dữ liệu quan hệ

* ***Sơ đồ ERD mức logic (ERD logic)***
* USER(**user\_id**, created\_date, modified\_date, display\_name, password, email, enable, provider, provider\_user\_id, secret, using2fa)
* ROLE(**role\_id**, name)
* USER\_ROLE(**user\_id, role\_id**)
* ADDRESS(**address\_id**, name)
* STORE(**store\_id**, name, email, phone, address\_detail, address, image)
* PRODUCT(**product\_id**, name, description, inventory, price, category, size, store, image)
* SIZE(**size\_id**, size)
* CATEGORY(**category\_id**, name, is\_activated)
* STAFF(**staff\_id**, name, email, phone, staff\_group, store, image)
* STAFF\_GROUP(**staff\_group\_id**, is\_activated, name)
* TIME\_KEEPING(**time\_keeping\_id**, month, num\_of\_shift, staff, salary, created\_date)
* SALARY(**salary\_id**, name, basic\_salary)



HInh 2. Sơ đồ dữ liệu quan hệ (ERD mức logic)

* ***Sơ đồ ERD mức vật lý (ERD vật lý)***



HInh 2. Sơ đồ dữ liệu quan hệ (ERD mức vật lý)

2.4 Cơ sở dữ liệu

* Bảng ADDRESS

Bảng 2. Bảng ADDRESS

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| Address\_id | Mã khu vực | bigint (20) | Khóa chính |
| name | Tên khu vực | Nvarchar (255) |  |

* Bảng CATEGORY

Bảng 2. Bảng CATEGORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| Category\_id | Mã loại sản phẩm | bigint (20) | Khóa chính |
| name | Tên loại sản phẩm | Varchar (255) |  |
| Is\_activated | Trạng thái kích hoạt | bit |  |

* Bảng SIZE

Bảng 2. Bảng SIZE

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| Size\_id | Mã kích cỡ | Bigint(20) | Khóa chính |
| size | Kích cỡ sản phẩm | varchar (255) |  |

* Bảng PRODUCT

Bảng 2. Bảng PRODUCT

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| Product\_id | Mã sản phẩm | bigint (20) | Khóa chính |
| description | Mô tả | varchar (255) |  |
| Inventory | Số lượng hiện có | Int |  |
| Name | Tên sản phẩm | varchar (255) |  |
| Price | Đơn giá | Double |  |
| Category | Mã loại sản phẩm | Bigint(20) | Khoá ngoại |
| Store | Mã cửa hàng | Bigint(20) | Khoá ngoại |
| size | Mã kích cỡ | Bigint(200 | Khoá ngoại |
| image | Ảnh sản phẩm | varchar (200) |  |

* Bảng ROLE

Bảng 2. Bảng ROLE

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| Role\_id | Mã vai trò | Bigint(20) | Khóa chính |
| name | Tên vài trò | Vachar(255) |  |

* Bảng USER\_ROLE

Bảng 2. Bảng USER\_ROLE

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| User\_id | Mã người dùng | Bigint(20) | Khóa chính,  Khoá ngoại |
| Role\_role | Mã vai trò | Bigint(20) | Khóa chính,  Khoá ngoại |

* Bảng USER

Bảng 2. Bảng USER

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| User\_id | Mã người dùng | Bigint(20) | Khóa chính |
| Created\_date | Ngày tạo | datetime |  |
| Display\_name | Tên hiển thị | Varchar(255) |  |
| Email | Email | Varchar(255) |  |
| Enabled | Trạng thái kích hoạt | Bit(1) |  |
| Modified\_date | Ngày xác thực | datetime |  |
| password | Mật khẩu | Varchar(255) |  |
| Provider | Loại xác thực | Varchar(255) |  |
| Provider\_user\_id | Mã loại xác thưc | Varchar(255) |  |
| Secret | Mã bảo mật | Varchar(255) |  |
| Using2fa | Dùng xác thực 2 bước 2FA | Bit(1) |  |

* Bảng STAFF

Bảng 2. Bảng STAFF

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| Staff\_id | Mã nhân viên | Bigint(20) | Khóa chính |
| Phone | Số điện thoại | Varchar (255) |  |
| Email | Email | Varchar (255) |  |
| Staff\_group | Mã loại nhân viên | Bigint(20) | Khoá ngoại |
| Store | Mã cửa hàng | Bigint(20) | Khoá ngoại |
| image | ảnh nhân viên | Varchar (255) |  |

* Bảng STAFF\_\_GROUP

Bảng 2. Bảng STAFF\_GROUP

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| Staff\_group\_id | Mã loại nhân viên | Bigint(20) | Khóa chính |
| Name | Tên loại nhân viên | varchar (255) |  |
| Is\_activated | Trạng thái kích hoạt | Bit(1) |  |

* Bảng SALARY

Bảng 2. Bảng SALARY

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| Salary\_id | Mã mức lương | Bigint(20) | Khóa chính |
| Name | Tên mức lương | Varchar (255) |  |
| Basic\_salary | Lương cơ bản | Varchar (255) |  |

* Bảng STORE

Bảng 2. Bảng STORE

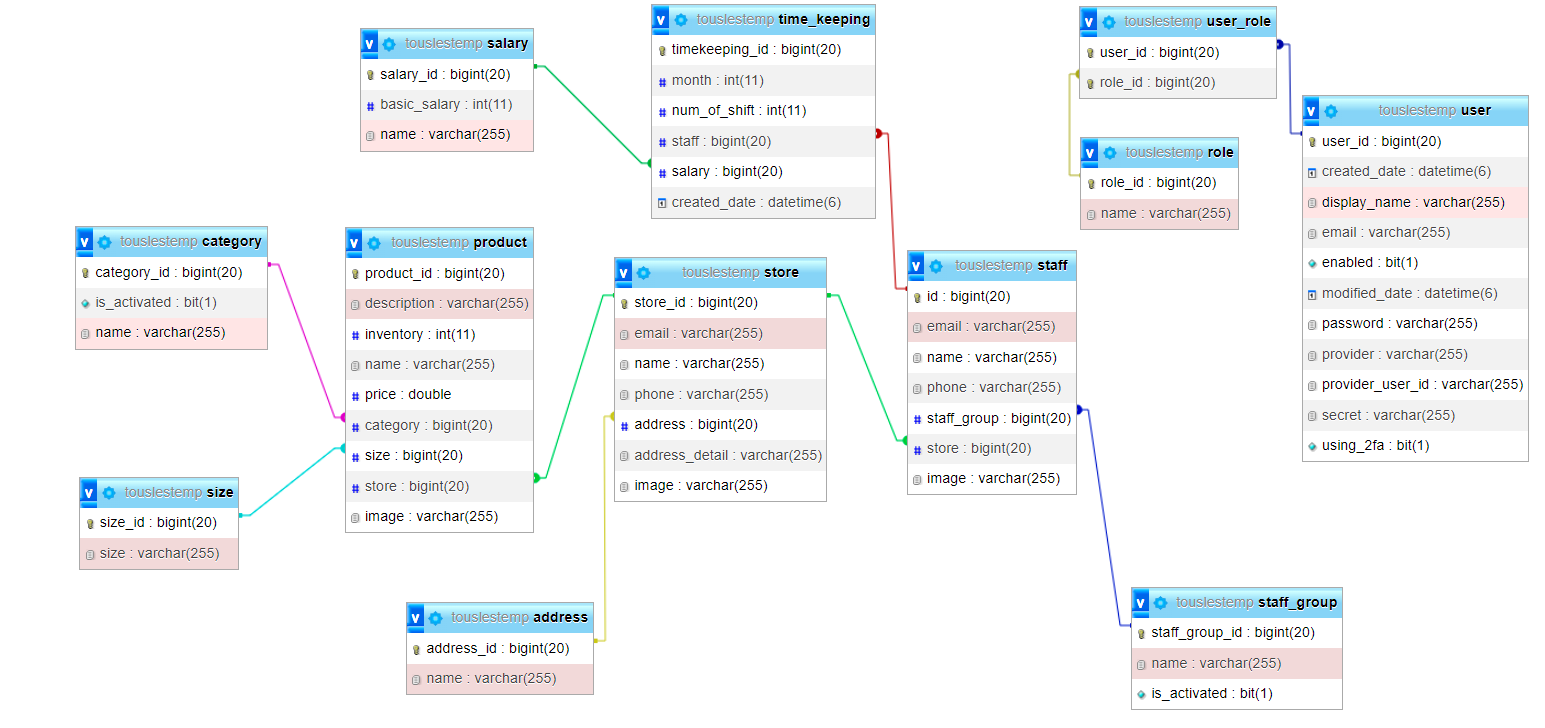
|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| Store\_id | Mã cửa hàng | Bigint(20) | Khóa chính |
| Phone | Số điện thoại | Varchar (255) |  |
| Email | Email | Varchar (255) |  |
| Address | Mã khu vực | Bigint(20) | Khoá ngoại |
| Address\_detail | Địa chỉ chi tiết | Varchar (255) |  |
| image | ảnh cửa hàng | Varchar (255) |  |

* Bảng TIME\_KEEPING

Bảng 2. Bảng TIME\_KEEPING

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Ý nghĩa** | **Kiểu dữ liệu** | **Ràng buộc** |
| MaLuong | Mã lương | Varchar (6) | Khóa chính |
| LuongCB | Lương cơ bản | Int |  |
| NgayDieuChinh | Ngày điều chỉnh | Date |  |
| MaNV | Mã nhân viên | Varchar (6) | Khóa ngoại |

* ***Lược đồ CSDL trên SQL Server***



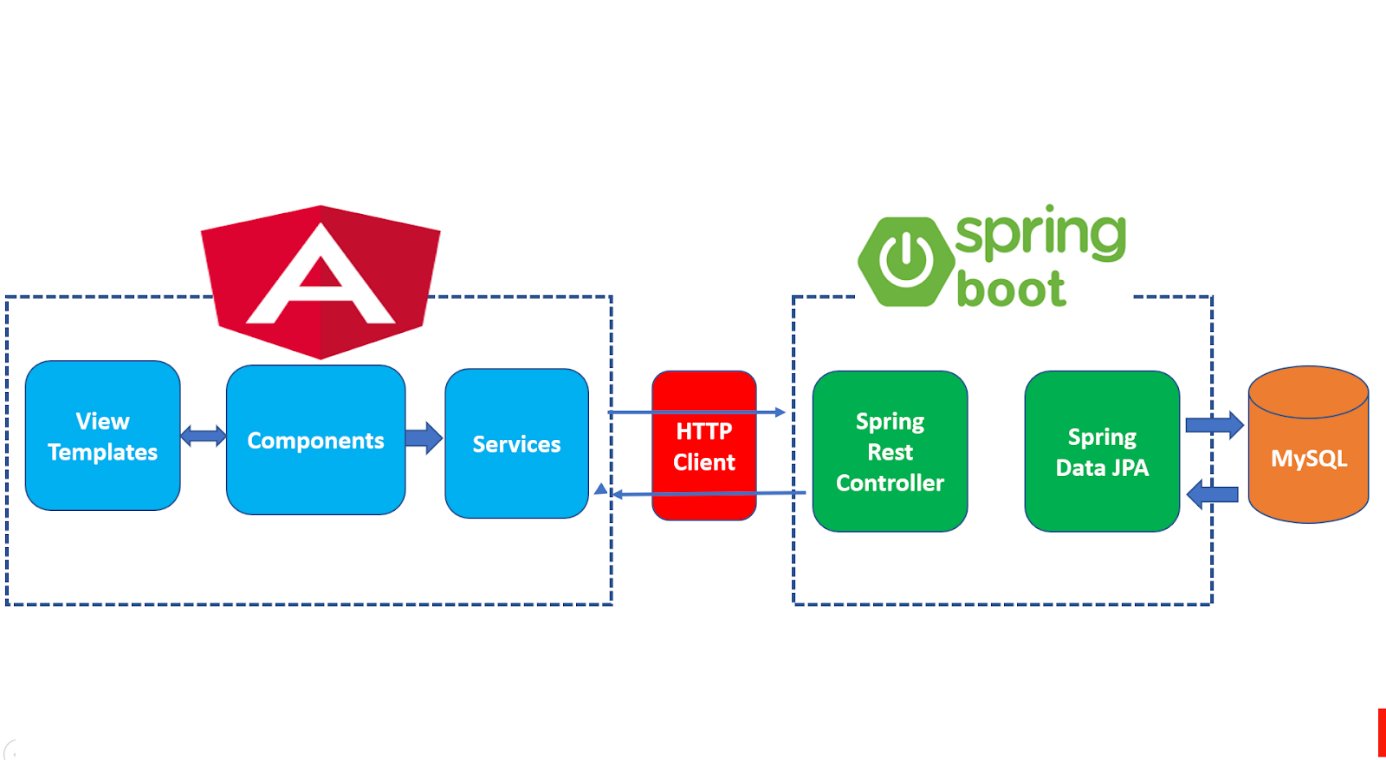
HInh 2. Lược đồ CSDL trên MySQL - Database Diagram

CHƯƠNG 3: LẬP TRÌNH ỨNG DỤNG

3.1 Cấu trúc hệ thống

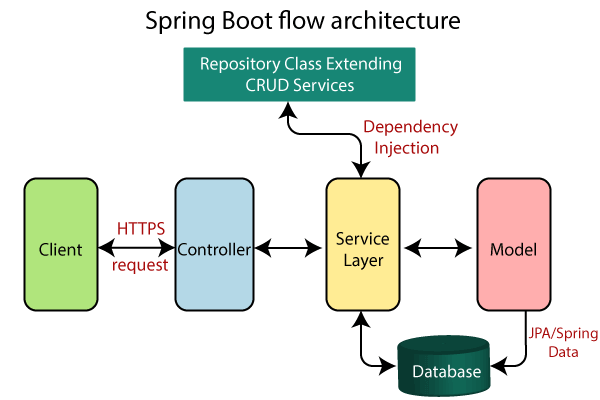
Ứng dụng Spring áp dụng cho Tous Les Temp là một ứng dụng web sử dụng Angular, Spring Boot, hệ quản trị cơ sở dữ liệu MySQL. Sự kết hợp giữa Spring Boot và Angular là phương án tối ưu trong việc xây dụng ứng dụng web. Lợi ích chính khi sử dụng Spring Boot cùng Angular cho đồ án này như sau:

* Tách biệt phần backend và frontend: Spring Boot là một framework phát triển phần backend mạnh mẽ, trong khi Angular là một framework phát triển phần frontend tiên tiến. Kết hợp cả hai giúp tách biệt rõ ràng giữa phần backend và frontend của ứng dụng, tạo điều kiện thuận lợi cho việc phát triển dự án
* Tích hợp dễ dàng: Spring Boot cung cấp các tính năng và công cụ để phát triển và triển khai ứng dụng backend. Angular, với mô hình single-page application, làm cho việc phát triển giao diện người dùng trở nên dễ dàng và linh hoạt, có khả năng tích hợp tốt với nhau thông qua các API và giao thức chuẩn.
* Hiệu suất cao: Angular cung cấp khả năng tạo ra các ứng dụng web động, tương tác mượt mà và có hiệu suất cao. Spring Boot, với tính năng tự động cấu hình, giúp xây dựng các ứng dụng backend chạy nhanh và ổn định.
* Quản lý dễ dàng: Spring Boot cung cấp cơ chế quản lý dependency và cấu hình linh hoạt thông qua Spring Boot Starter và Angular qua Angular CLI

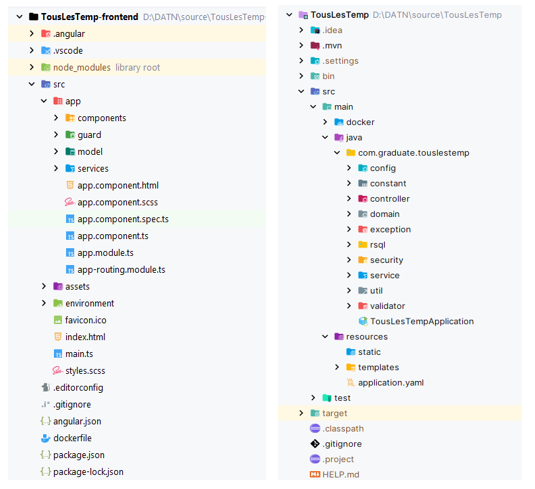


Hình 3. Mô hình kết hợp Spring và Angular

Dữ liệu phía server của hệ thống sẽ tuần tự thực hiện như sau

1. Yêu cầu (Request): Yêu cầu từ phía client được gửi đến ứng dụng Spring thông qua các giao thức như HTTP hoặc WebSocket.
2. Tiền xử lý (Pre-processing): Ứng dụng Spring sẽ tiếp nhận yêu cầu và thực hiện các xử lý tiền xử lý cần thiết, bao gồm kiểm tra và xác thực yêu cầu, kiểm tra quyền truy cập, và các xử lý khác tùy thuộc vào logic kinh doanh của ứng dụng.
3. Xử lý (Processing): Sau khi tiền xử lý hoàn tất, ứng dụng Spring sẽ xử lý yêu cầu dựa trên các controller hoặc endpoints được định nghĩa..
4. Gọi dịch vụ (Service Invocation): Trong quá trình xử lý yêu cầu, ứng dụng Spring có thể gọi các dịch vụ khác như các service bean.
5. Truy xuất cơ sở dữ liệu (Database Access): Nếu cần, ứng dụng Spring có thể thao tác với CSDL để lưu giữ hoặc truy cập đến dữ liệu thông qua Spring Data JPA hoặc các công nghệ truy vấn dữ liệu khác.
6. Xử lý kết quả (Result Processing): Sau khi hoàn thành xử lý yêu cầu và tương tác với dữ liệu, ứng dụng Spring trả về kết quả hoặc dữ liệu tương ứng cho client. Kết quả có thể là các dữ liệu JSON, XML, hoặc HTML
7. Trả lời (Response): Kết quả được đóng gói trong phản hồi (response) và gửi trả về cho client qua giao thức tương ứng

Hình 3. Luồng dữ liệu phía backend

Cấu trúc source file dự án

Hình 3. Cấu trúc source file

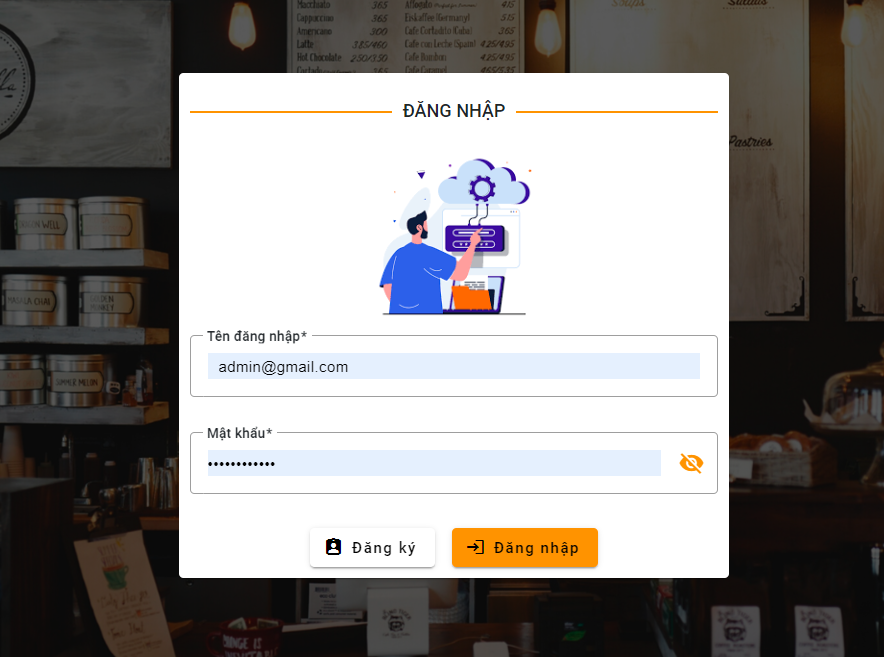
3.2 Các giao diện trong hệ thống

3.2.1 Giao diện đăng ký, đăng nhập và xác thực 2FA

* **Đăng nhập**

Giao diện đăng nhập, người dùng thường được yêu cầu cung cấp thông tin đăng nhập để xác thực danh tính của họ, người quản lý sẽ tiến hành cung cấp thông tin đăng nhập để truy cập vào hệ thống. Sau khi đã nhập đủ thông tin tên đăng nhập và mật khẩu, nhấn nút “Đăng nhập” để tiến hành xác thực.

Xử lý thông tin đăng nhập: Ứng dụng kiểm tra thông tin được cung cấp và so sánh chúng với cơ sở dữ liệu người dùng đã đăng ký trước đó. Nếu thông tin đúng, người dùng sẽ được chuyển hướng đến trang chính của hệ thống. Nếu thông tin không chính xác, ứng dụng sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin.

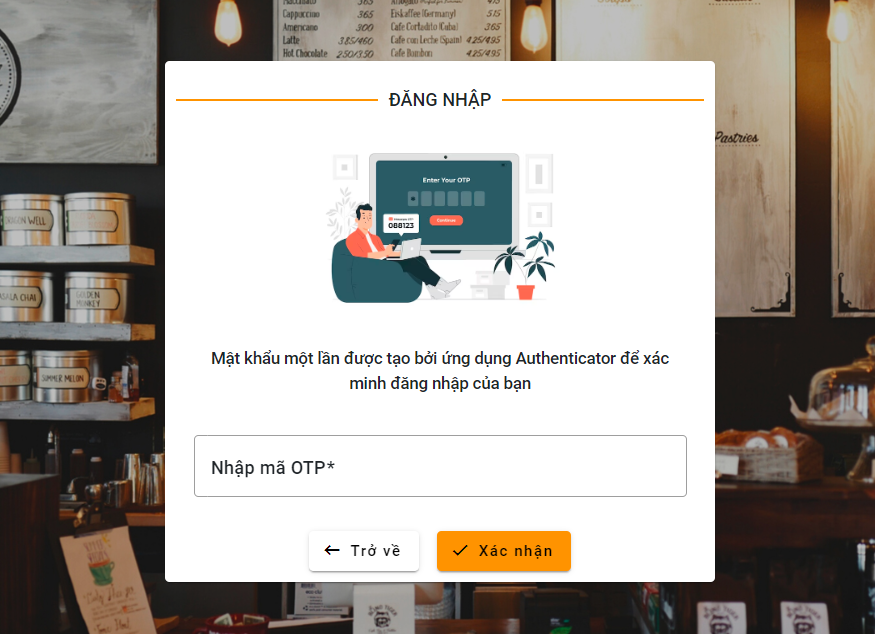


Hình 3. Đăng nhập hệ thống

* **Xác thực OTP**

Nếu trước đó, tài khoản được đăng ký và tích chọn sử dụng xác thực hai yếu tố 2FA, sau khi kiểm tra thông tin đăng nhâp hợp lệ, sẽ được yêu cầu cung cấp mã OTP để thực hiện xác thực hai yếu tố. Sau khi người dùng đã nhập mã OTP, họ sẽ nhấn vào nút "Xác nhận" để tiến hành xác thực.

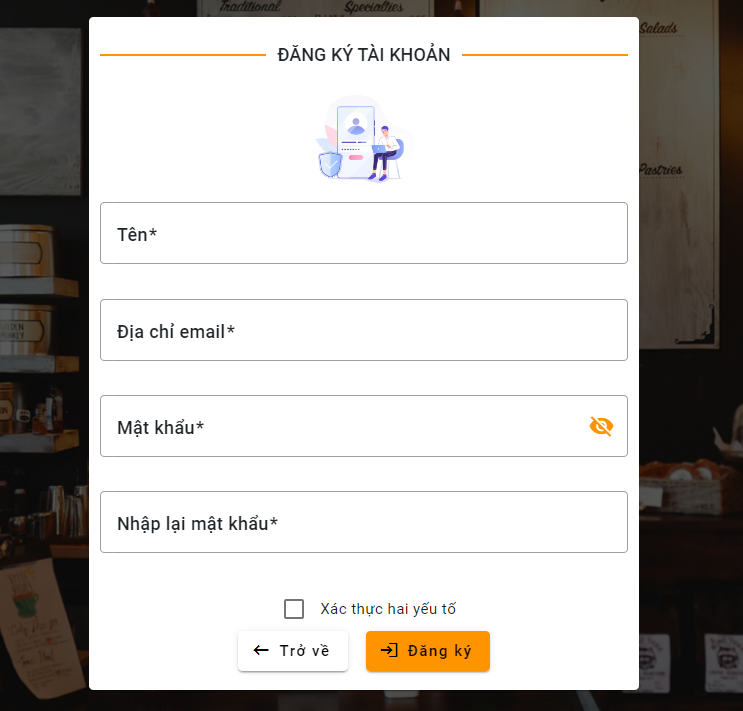
Quá trình xử lý OTP: Ứng dụng sẽ kiểm tra mã OTP được nhập và so sánh nó với mã OTP hợp lệ. Nếu mã OTP đúng, người dùng sẽ được chuyển hướng đến trang chủ hệ thống. Nếu mã OTP không chính xác hoặc hết hạn, giao diện sẽ hiển thị thông báo lỗi và yêu cầu người dùng nhập lại mã OTP.



Hình 3. Xác thực OTP

* **Đăng ký**

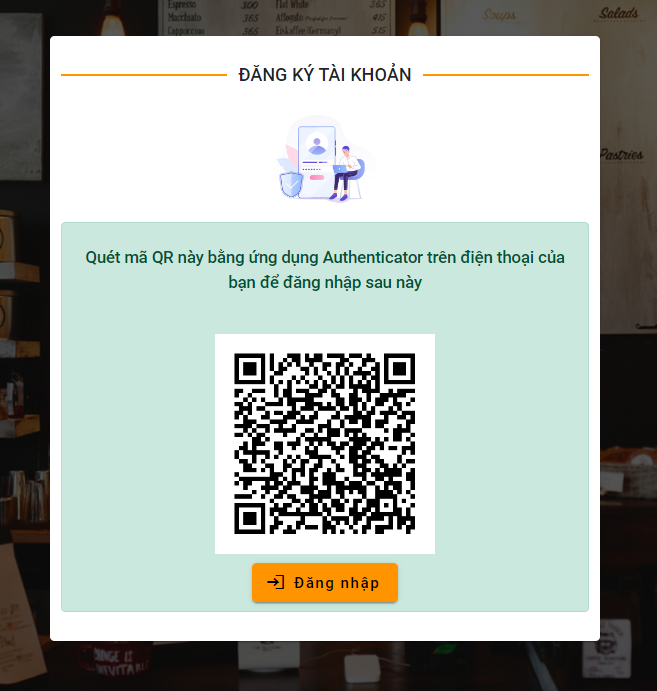
Trường hợp người quản lý muốn tạo tài khoản mới thì thực hiện nhấn nút   
“Đăng ký” để tiến hành đăng ký tài khoản. Giao diện đăng ký thường bao gồm các trường văn bản để người dùng nhập thông tin cá nhân như tên, địa chỉ email, và mật khẩu. Nhấn nút “Đăng ký” để tiến hành đăng ký.

Xác thực thông tin: Ứng dụng có thể kiểm tra tính hợp lệ của thông tin đăng ký và gửi mail đăng ký thành công đến email được đăng ký. Xử lý thông tin đăng ký: Ứng dụng sẽ xử lý thông tin đăng ký người dùng đã cung cấp. Điều này bao gồm việc kiểm tra tính hợp lệ của thông tin, tạo tài khoản mới và lưu thông tin vào cơ sở dữ liệu.

Hình 3. Đăng ký

* Xác nhận đăng ký thành công có sử dụng 2FA

Khi thực hiện đăng ký thành công, kiểm tra xem nếu tài khoản khi đăng ký có kích hoạt xác thực hai yếu tố, thì hệ thống sẽ hiển thị giao diện đăng ký thành công, kèm theo đó là một mã QR code được sinh ra để thực hiện quét mã QR và nhận OTP ở ứng dụng Authenticator mỗi lần đăng nhập sau này. Quá trình này được thực hiện bởi Spring Security Framework phía server

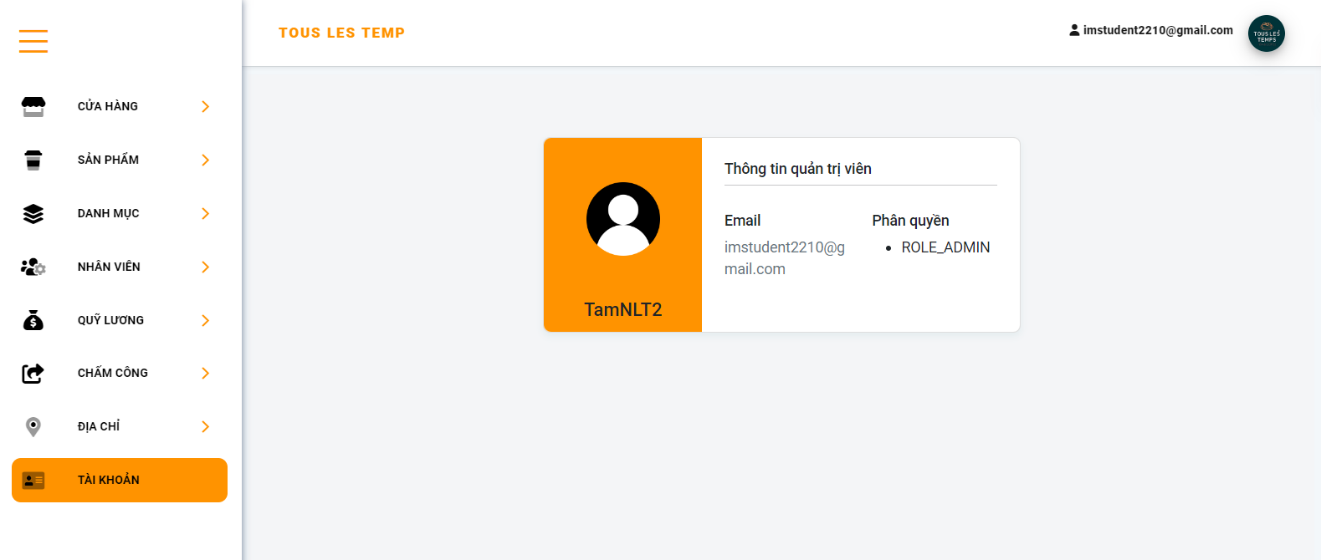


Hình 3. Đăng ký thành công có sử dụng 2FA

* Xác nhận đăng ký thành công không sử dụng 2FA



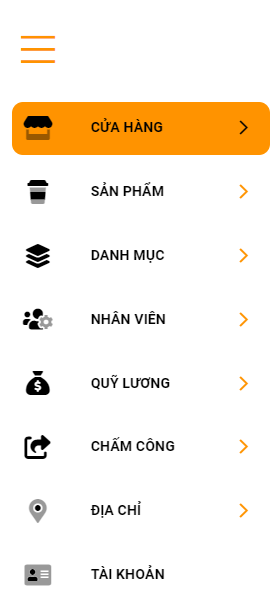
Hình 3. Đăng ký thành công không sử dụng 2FA

Từ đây, người dùng có thể thực hiện chọn nút “Đăng nhập” để đến với giao diên đăng nhập và đăng nhập hệ thống với tài khoản vừa tạo.

Hình 3. Thông tin tài khoản

3.2.2 Thanh menu hệ thống

Bước đầu người sử dụng có thể tìm được nội dung cần xử lý thao tác. Mỗi mục sẽ có một menu con được tự động thu phóng tương ứng với mục đang được thao tác



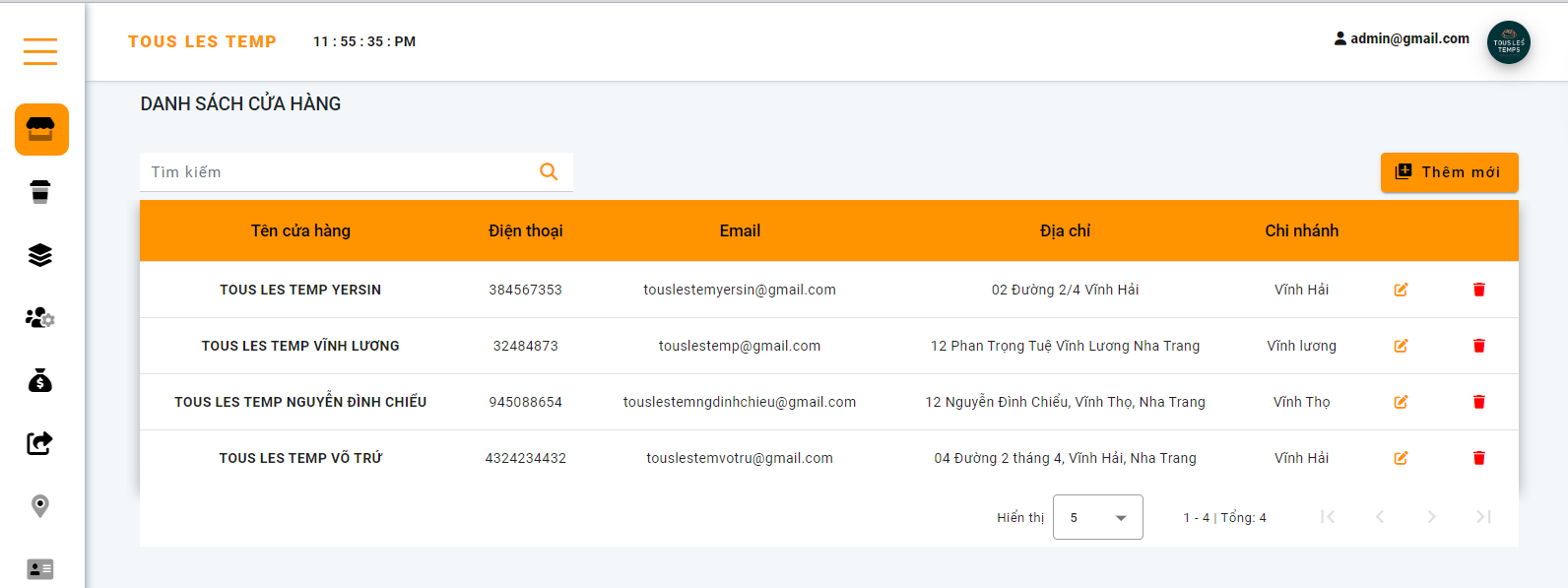
Hình 3. Dashboard của hệ thống

3.2.3 Giao diện quản lý cửa hàng

* Thông danh sách cửa hàng

Thông tin về danh sách cửa hàng của hệ thống sẽ hiển thị ở đây dưới dạng bảng, gồm các thông tin của cửa hàng nhue tên, địa chỉ, email, điện thoại..

Bên cạnh đó ta có thể xem thông tin chi tiết từng cửa hàng thông qua việc click chọn tên của cửa hàng trong bảng, hệ thống sẽ thực hiện điều hướng sang trang cập nhật thông tin của hàng. Quá trình này được Angular router thực hiện điều hướng đính kèm theo ID của cửa hàng được click chọn, từ đó sẽ xử lí gọi đến storeService và sử dụng giao thức http để gọi API **getStoreById** của server. Dữ liệu sẽ được gửi vể client và hiển thị lên giao diện. Ở mỗi dòng của bảng sẽ có 2 button đê thực hiện chức năng Cập nhật và Xoá cửa hàng.

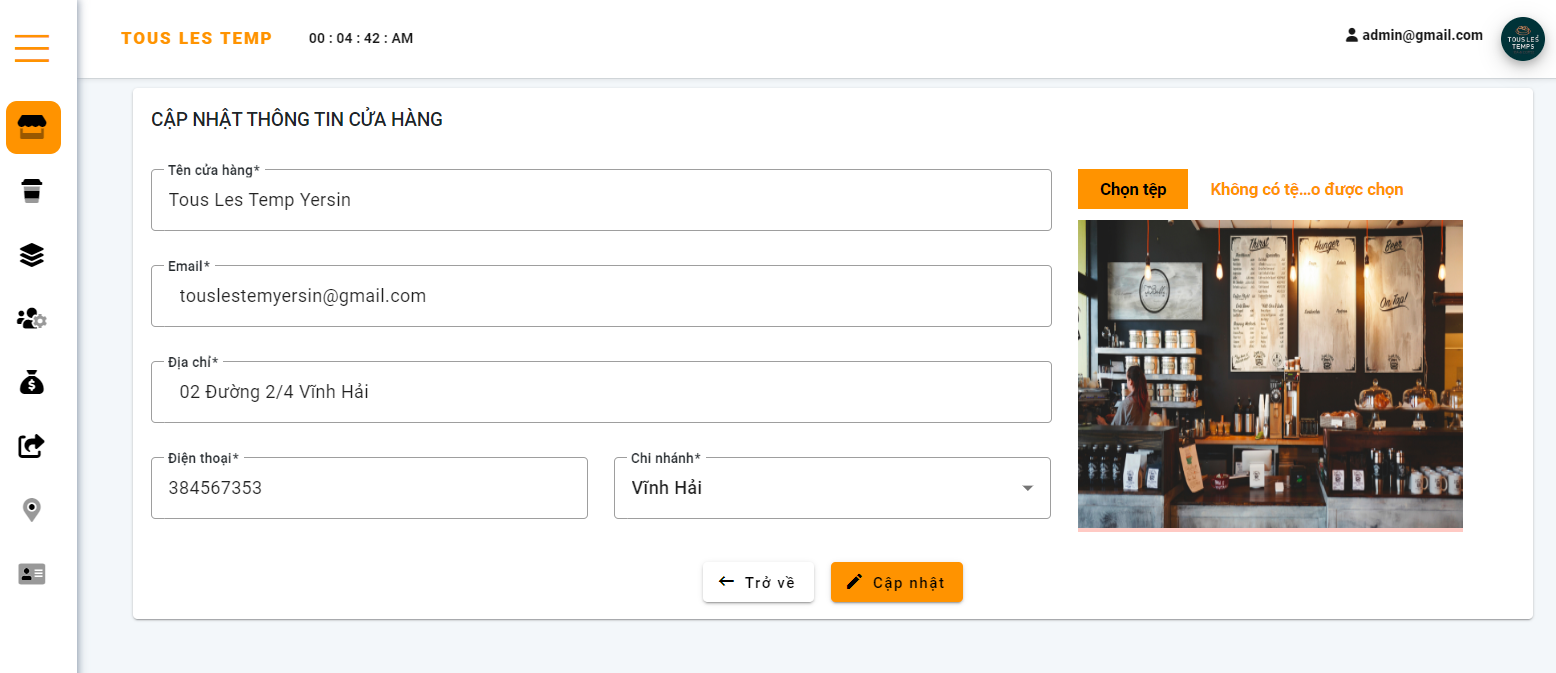


Hình 3. Danh sách cửa hàng

Với việc cập nhật thông tin cửa hàng, khi người quản lý click chọn icon cập nhật sẽ tiến hàng thực hiện gọi đến trang Cập nhật, hệ thống sẽ dựa vào ID của cửa hàng được chọn mà thực hiện hiện thị lại thông tin của cửa hàng đó, tiến hành thay đổi các thông tin và cập nhật hình ảnh nếu cần. Khi đã cập nhật xong đầy đủ các thông tin, các trường dữ liệu sẽ được kiểm tra đã hợp lệ hay chưa và người dùng sẽ click button   
Cập nhật để gọi thực hiện API **updateStore** và hoàn tất việc cập nhật sẽ hiển thị trở lại trang danh sách cửa hàng đông thời có một thông báo cập nhật thành công xuất hiện

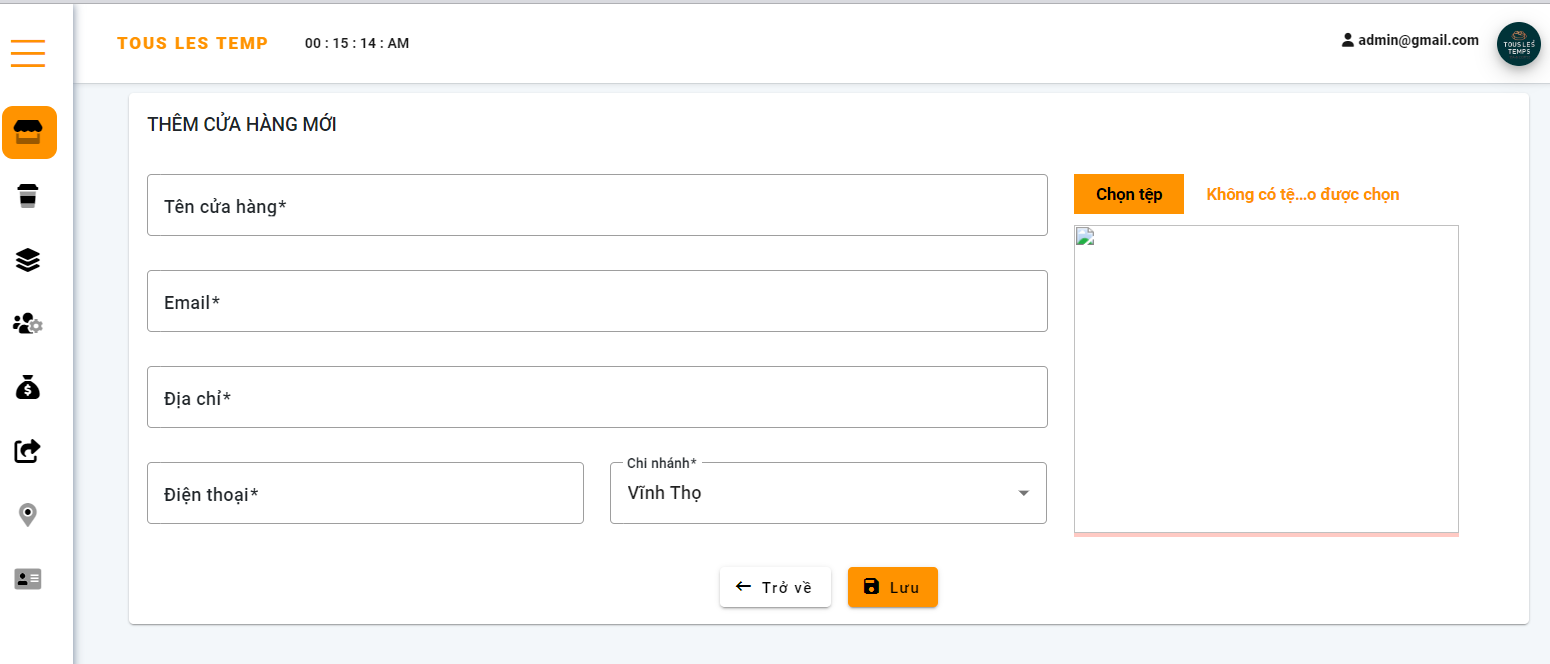
Với việc xoá cửa hàng, khi click button Xoá, phía server sẽ thực hiện API deleteStore, hiển thị thông báo xoá thành công và hiển thị danh sách cửa hàng mới

Ở màn hình danh sách cửa hàng ta cũng có button Thêm mới để thêm 1 cửa hàng mới vào hệ thống. Một cửa hàng sẽ được thêm mới và hệ thống nếu như tên của nó chưa tồn tại trong danh sách cửa hàng. Phía server sẽ thực hiện một API khác để lấy được   
danh sách các khu vực và đẩy dữ liệu đó vào một combo box có thể dễ dàng lựa chọn



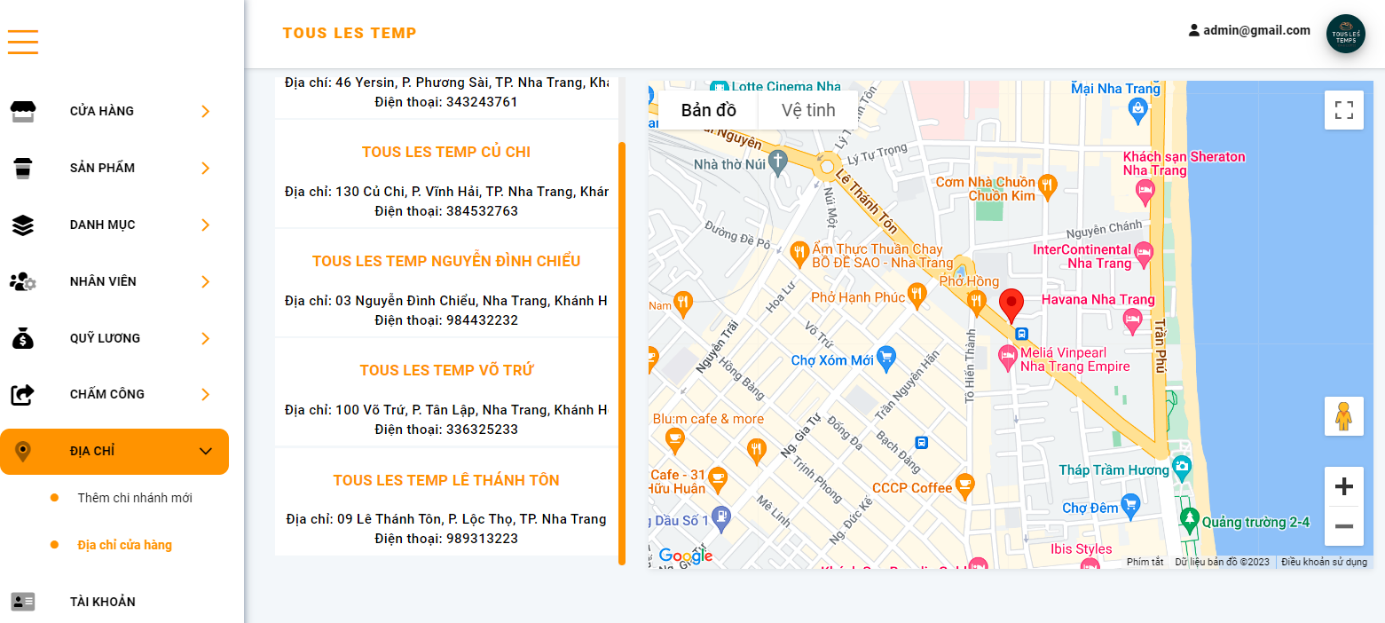
Hình 3. Cập nhật thông tin cửa hàng

Việc hiển thị ảnh sẽ được Angular xử lí. Ảnh chọn từ đây sẽ được xử lý và lưu trữ trên firebase, lấy ra thông tin đường dẫn của ảnh để lưu giữ vào CSDL, nhờ đó CSDL sẽ không bị “phình to” khi ta lưu trữ thông tin một ảnh bằng byte nhờ vào cơ chế của lớp **MultipartFile** trong Spring Boot



Hình 3. Thêm mới cửa hàng

* Địa chỉ các cửa hàng thông qua Google Map API



Hình 3. Đỉa chỉ các cửa hàng thông qua Google Map API

Nhờ việc sử dụng công nghệ Maps JavaScript API: API này cho phép ta nhúng bản đồ tương tác vào trang web

Đồng thời Geocoding API từ công nghệ này của Google cho phép ta chuyển đổi địa chỉ thành tọa độ địa lý (vĩ độ và kinh độ) và ngược lại, giúp dễ dàng tìm kiếm

Việc thực hiện API của Google đòi hỏi ta phải tạo một ứng dụng API từ Google Map Platform, từ đó lấy được API key và sử dụng cho trang web của mình.

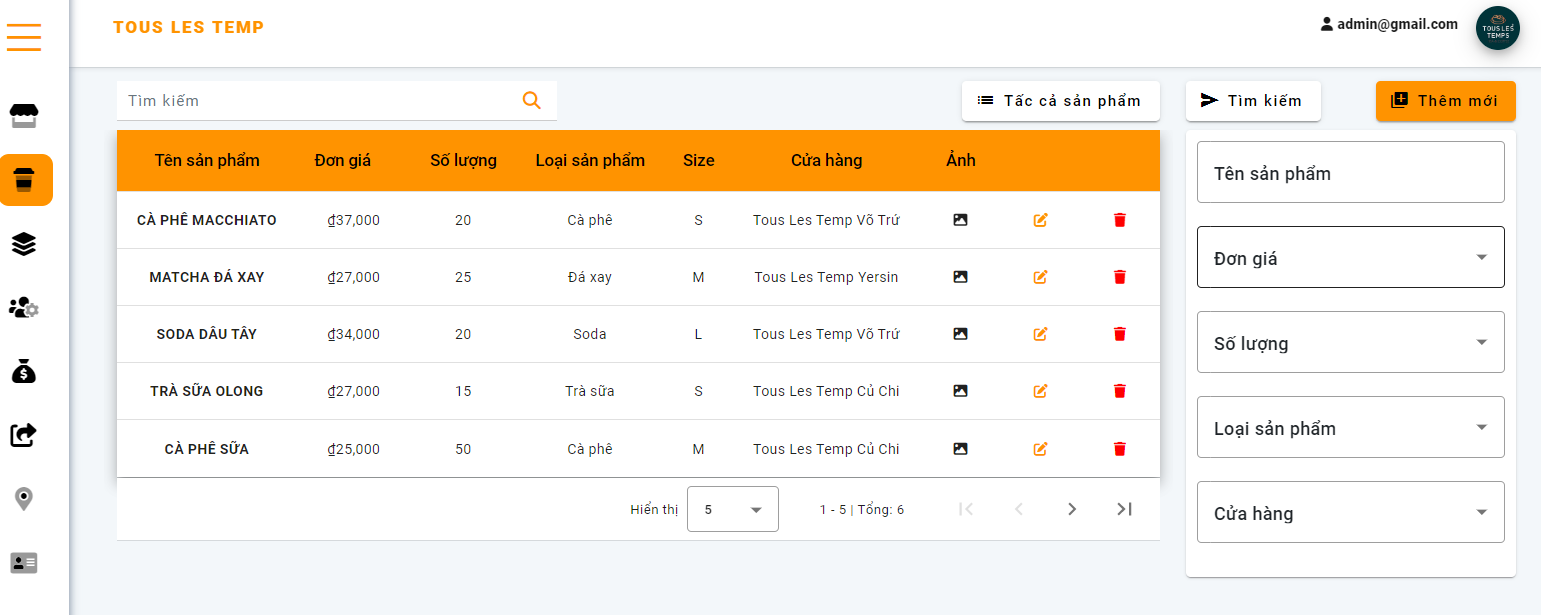
Ở giao diện này, khi ta click vào dòng chứa thông tin của cửa hàng từ danh sách bên trái, hệ thống sẽ thực hiện gọi đến service để thực thi hàm tìm kiếm đỉa chỉ **findStores(addressDetail)** với tham số truyền vào sẽ là địa chỉ cụ thể cửa mỗi cửa hàng, chuỗi địa chỉ này sẽ được chuyển đồi về kinh độ và vĩ độ và xử lý hiển thị địa chỉ của cửa hàng tương ứng. Nhờ vào giao diện này, mà người quản lý có thể quản lý dễ dàng địa chỉ từng cửa hàng của hệ thống, đồng thời sẽ phù hợp khi áp dụng trang web cho người dùng dễ dàng tìm kiếm cửa hàng mà mình cần đến.

3.2.4 Giao diện quản lý sản phẩm

* Thông tin sản phẩm

Giao diện quản lý danh sách sản phẩm của cả hệ thống, cung cấp thông tin của sản phẩm về đơn giá, loại, đang có trong menu của cửa hàng nào. Mỗi dòng của bảng đều có thêm ba nút để người dùng thao tác như xem ảnh sản phẩm, cập nhật cũng như xoá sản phẩm. Tìm kiếm bằng input để nhập giá trị tên sản phẩm, các giá trị gần đúng sẽ được Angular xử lý và hiển thị ngay sau khi nhập

Nhập những thông tin về tiêu chí tìm kiếm về sản phẩm, nhấn nút tìm kiếm, hệ thống sẽ thực hiện xử lí với câu query tương ứng được truyền xuống server, dữ liệu sẽ được tìm kiếm bằng RSQL (RESTful Service Query Language) là một ngôn ngữ truy vấn được sử dụng trong các dịch vụ RESTful để tìm kiếm và lọc dữ liệu. Nó cho phép xây dựng các truy vấn phức tạp để truy xuất dữ liệu từ nguồn dữ liệu RESTful một cách dễ dàng và mạnh mẽ.



Hình 3. Danh sách tấc cả sản phẩm

* Tạo mới sản phẩm



Hình 3. Tạo mới sản phẩm

Thực hiện chức năng thêm sản phẩm mới sản phẩm, kiếm soát các thông tin vào phải hợp lệ mới thực hiện submit form và gợi API thêm mới sản phẩm. Dách sách loại sản phẩm hiển thị ở combox là danh sách loại sản phẩm nhưng đang ở trạng thái “còn hàng” chứ không phải tấc cả loại sản phẩm đều được hiển thị. Khi chọn ảnh, thực hiện lưu trữ đường dẫn ảnh lên firebase. Khi thêm mới thành công, một toast menu thông báo thánh công sẽ hiển thị và thực hiển chuyển hướng về giao diện tấc cả sản phẩm

* Giao diện cập nhật thông tin sản phẩm

Khi thực hiện chuyển tiếp từ trang danh sách sản phẩm với ID của sản phẩm được chọn, ở giao diện cập nhật sẽ nhận thông tin sản phẩm hiện tại cần cập nhật bằng xử lý API **getProductById** ở backend, đồng thời hiển thỉ ảnh của sản phẩm. Việc cập nhật sẽ được thực hiện khi cung cấp đầy đủ các trường thông tin.

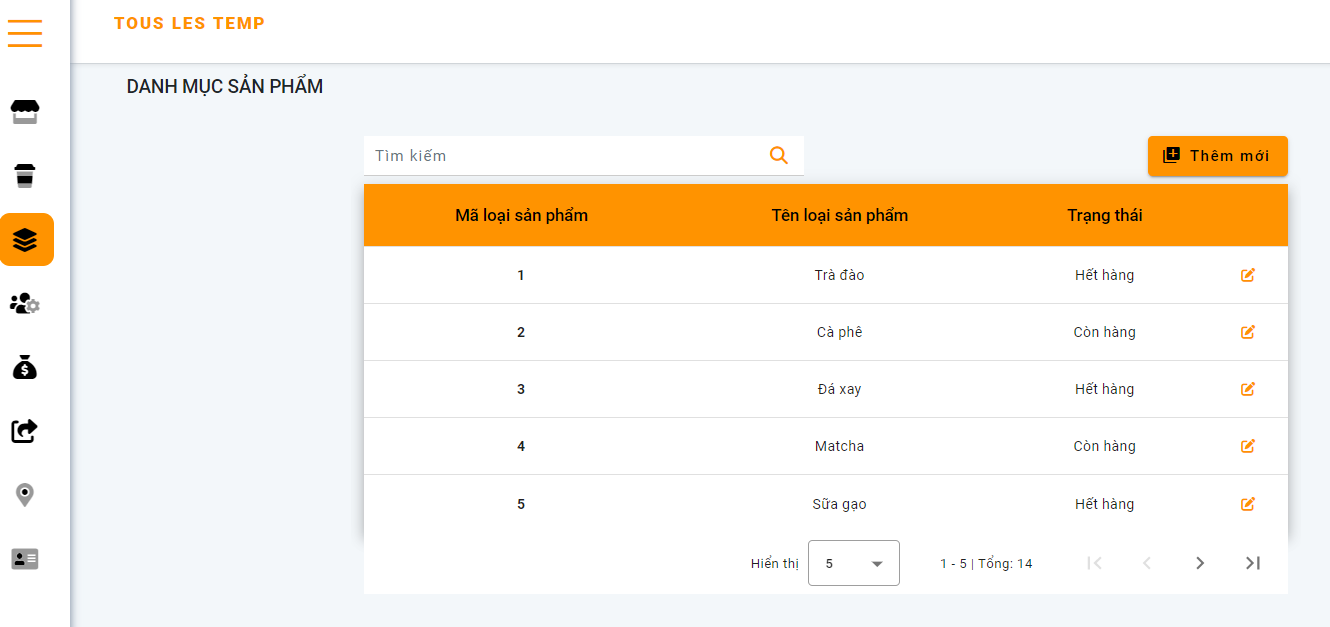
Ở xử lí này, hệ thống phát sinh vấn đề chưa được giải quyết là nếu cập nhật sản phẩm có tên trùng với một sản phẩm đã tồn tại nhưng khác cửa hàng thì vẫn thực hiện cập nhật thành công.



Hình 3. Cập nhật thông tin sản phẩm

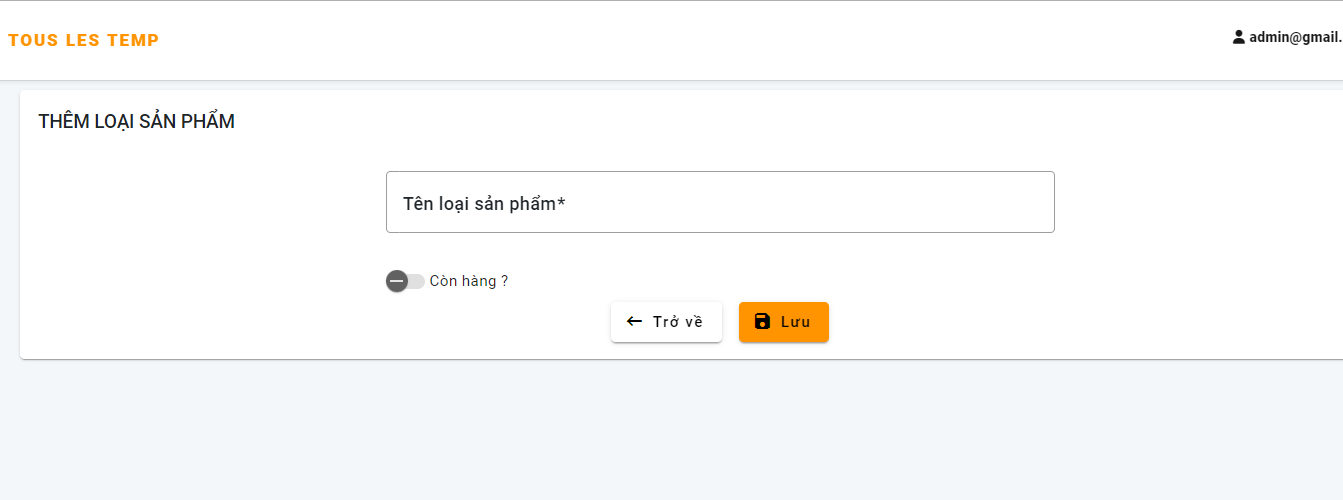
* Giao diện danh sách loại sản phẩm

Danh sách các loại sản phẩm trong menu từng cửa hàng sẽ hiển thị ở đây. Cũng như các màn hình trước, chức năng tìm kiếm, cập nhật và thêm mới cũng được xử lý. Số lượng loại sản phẩm nhiều sẽ thực hiện phân trang



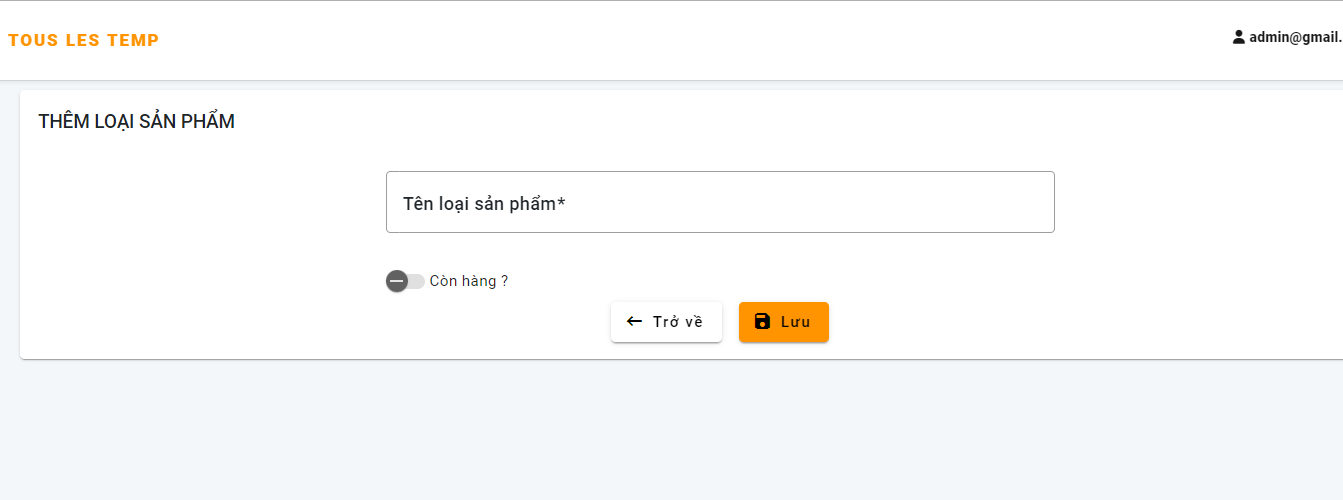
Hình 3. Danh mục loại sản phẩm

* Thêm mới loại sản phẩm



Hình 3. Thêm mới loại sản phẩm

* Cập nhật thông tin loại sản phẩm



Hình 3. Cập nhật loại sản phẩm

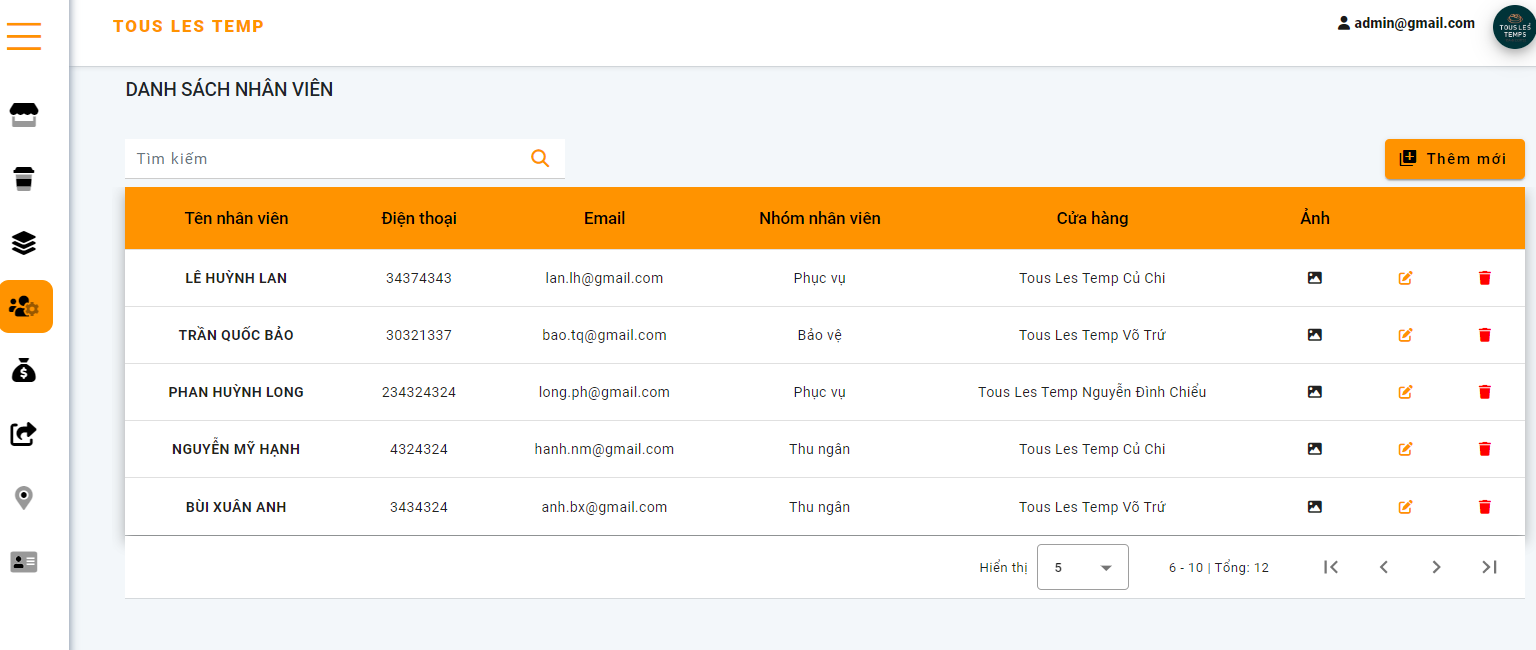
Thực hiện thêm mới và cập nhật thông tin của danh mục loại sản phẩm được thực hiện ở hai màn hình trên. Khi hoàn thành sẽ về trang danh sách loại sản phẩm

Khi thực hiện thao tác ở giao diện quản lý danh sách sản phẩm, người dùng có thể thực hiện tìm kiếm gần đúng bằng việc nhập tên nhân viên vào input tìm kiếm. Mặt khác khi người dùng muốn thực hiện tìm kiếm nhiều tiêu chí, họ sẽ điền những thông tin cần thiểt vào tiêu chí cần tìm, phía server sẽ dựa theo giá trị những input được nhập mà sinh ra câu query tương ứng và thực hiện gọi API tìm kiếm với ngôn ngữ RSQL. Kết quả tìm được sẽ hiển thị nhanh chóng lên màn hình, người dùng dễ dàng thao tác cập nhật, xoá

3.2.5 Giao diện quản lý nhân viên

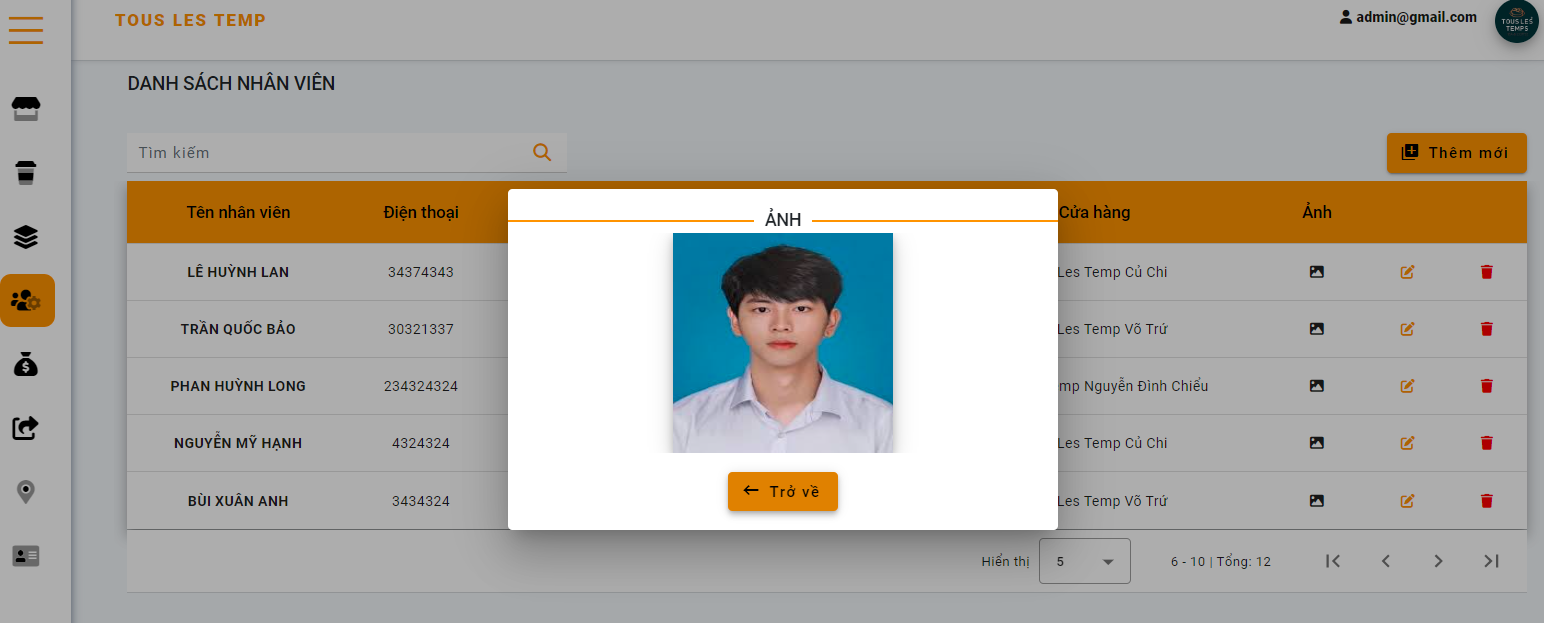
* Quản lý tấc cả nhân viên

Người dùng sẽ thấy tấc cả thông tin của nhân viên thông qua một bảng dữ liệu, bao gồm các thông tin về nhân viên như tên, email, điện thoại, vị trí làm việc, nơi làm việc. Chưc năng thêm mới, xoá cập nhật và tìm kiếm sẽ được thực thi từ màn hình này



Hình 3. Quản lý danh sách nhân viên

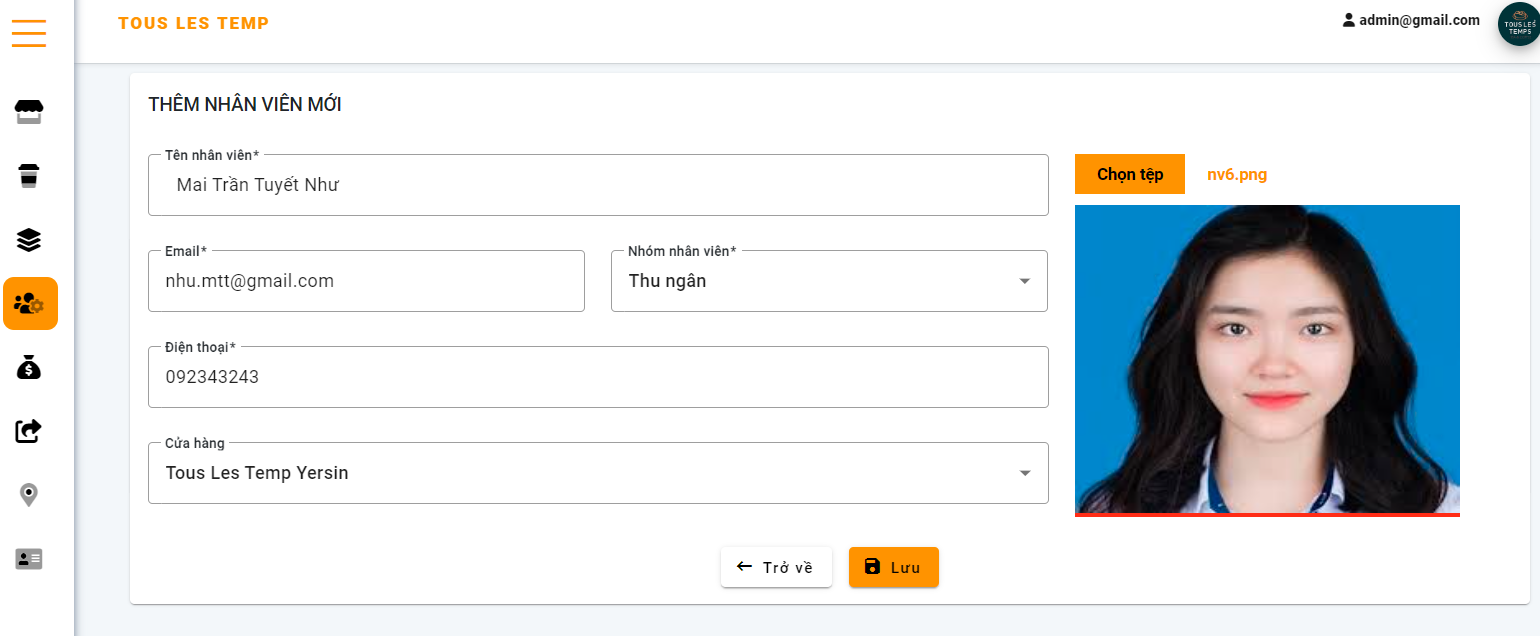
Khi click vào để xem ảnh của nhân viên, một popup sẽ xuất hiện và hiển thị hình ảnh nhân viên. Việc xử lí được Angular đảm nhiệm, lấy ID của nhân viên được chọn truyền qua poup này bằng cơ chế của **MatDialogModule** từ **Angular Material**



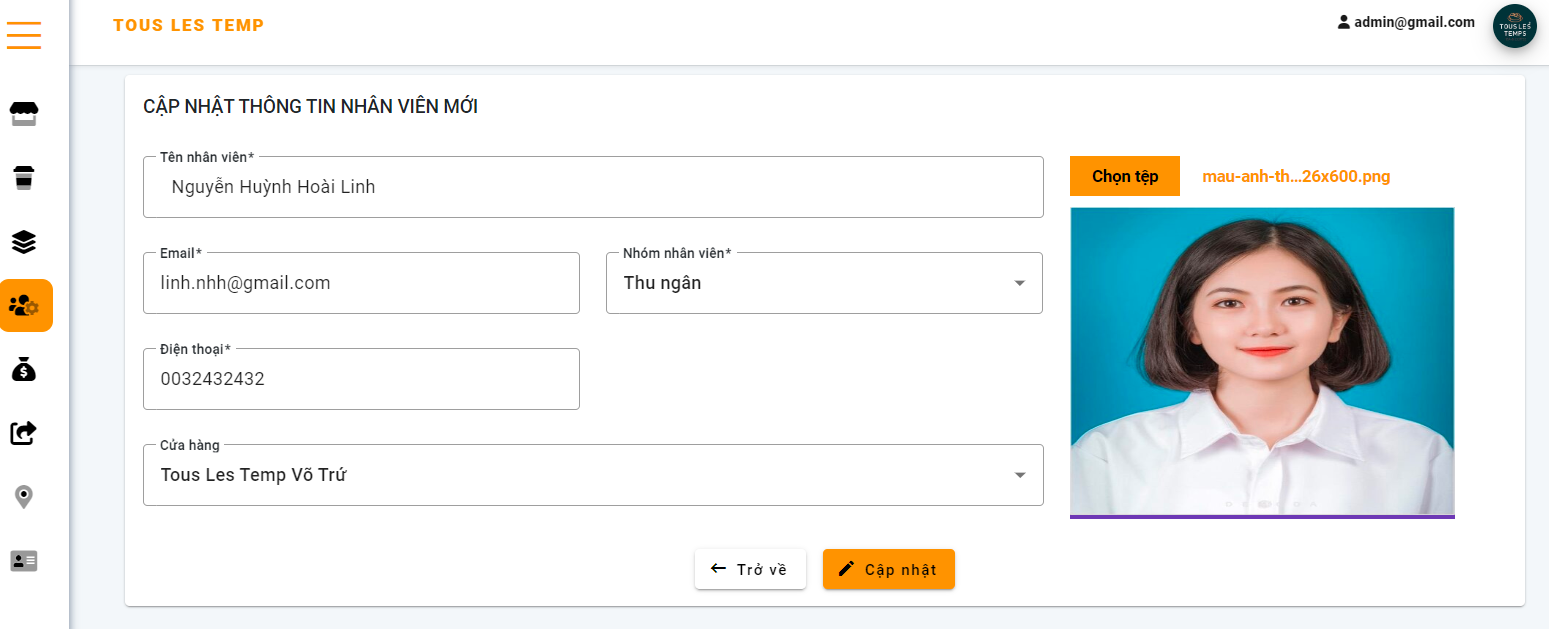
Hình 3. Dialog hiển thị ảnh nhân viên

* Thêm mới nhân viên

Thực hiện thêm mới một nhân viên vào hệ thống thông qua giao diện này, các thông tin đươc gọi API để hiển thị lên hai combobox là danh sách cửa hàng và danh sách vị trí công việc. Ở thao tác thêm mới, sẽ kiểm tra tên nhân viên có tồn tại trong hệ thống hay chưa tránh trường hợp bị trùng tên với nhau, khi đó hệ thống sẽ có thông báo để xử lý cảnh báo người dùng



Hình 3. Thêm mới nhân viên

* Cập nhật thông tin nhân viên

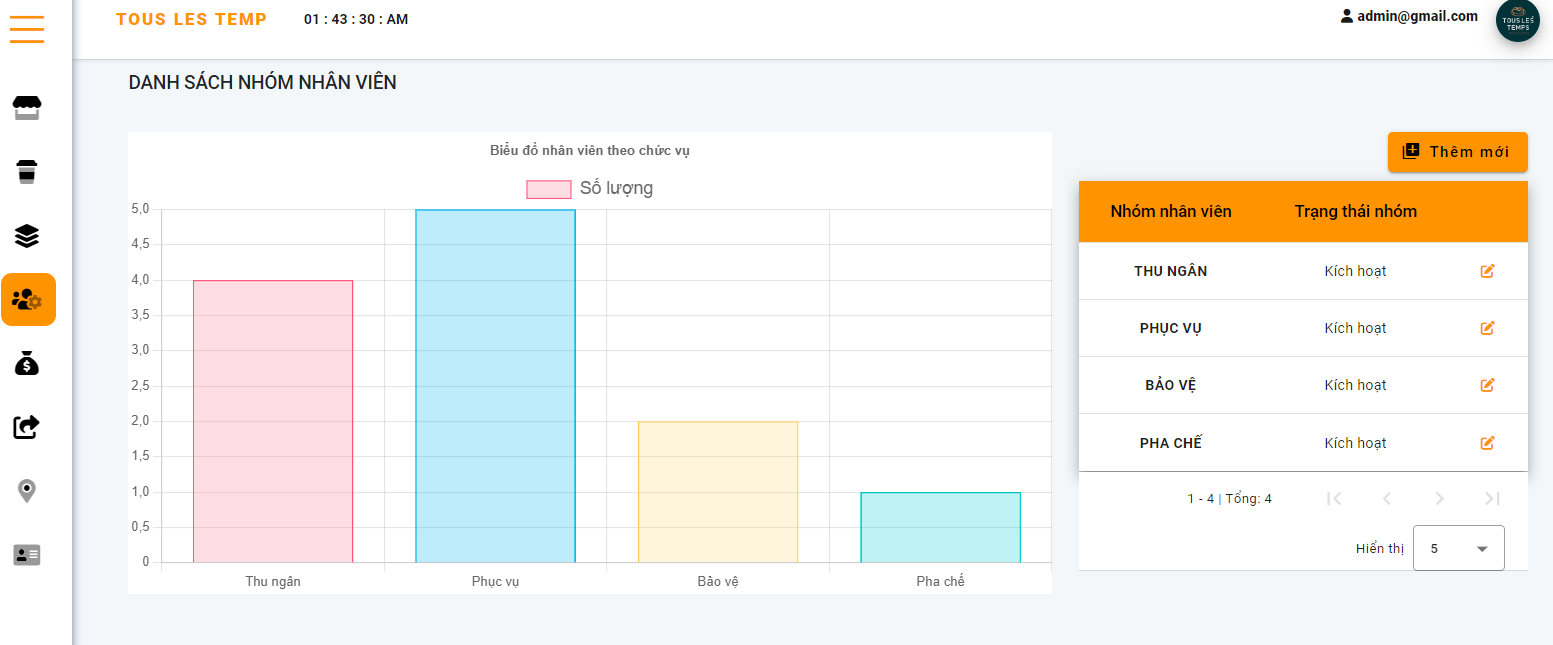
Hình 3. Cập nhật thông tin nhân viên

* Thông tin nhóm nhân viên

Ở giao diện này, bên cạnh các chức năng hiển thị dữ liệu dạng bảng, cập nhật, thêm mới, phân trang đều được triển khai, sử dụng biểu đồ để thực hiện thống kế số lượng nhân viên trong từng nhóm nhân viên.

Thống kê này là sự kết hợp chặt chẽ giữa cả hai phía backend lẫn frontend. Dùng Spring để tạo một Restful API để đếm số lượng nhân viên của từng nhóm, sau đó gửi dữ liệu ấy lên giao diện biểu đồ thống kê. Thực hiện được việc này là Angular đã nhờ vào Chart.js, để tạo và hiển thị các biểu đồ tương tác trực quan trong các ứng dụng web. Nó cung cấp một cách đơn giản để tạo ra các biểu đồ như đường, cột, tròn, với các tính năng linh hoạt và khả năng tùy chỉnh cao

Ở biểu đồ này, ta thấy rõ mỗi một cột sẽ đại diện cho một nhóm nhân viên ở chuỗi cửa hàng, việc thống kế sẽ dựa vào giá trị dữ liệu nhận được từ API backend mà sẽ có kích thước chiểu cao cột khác nhau

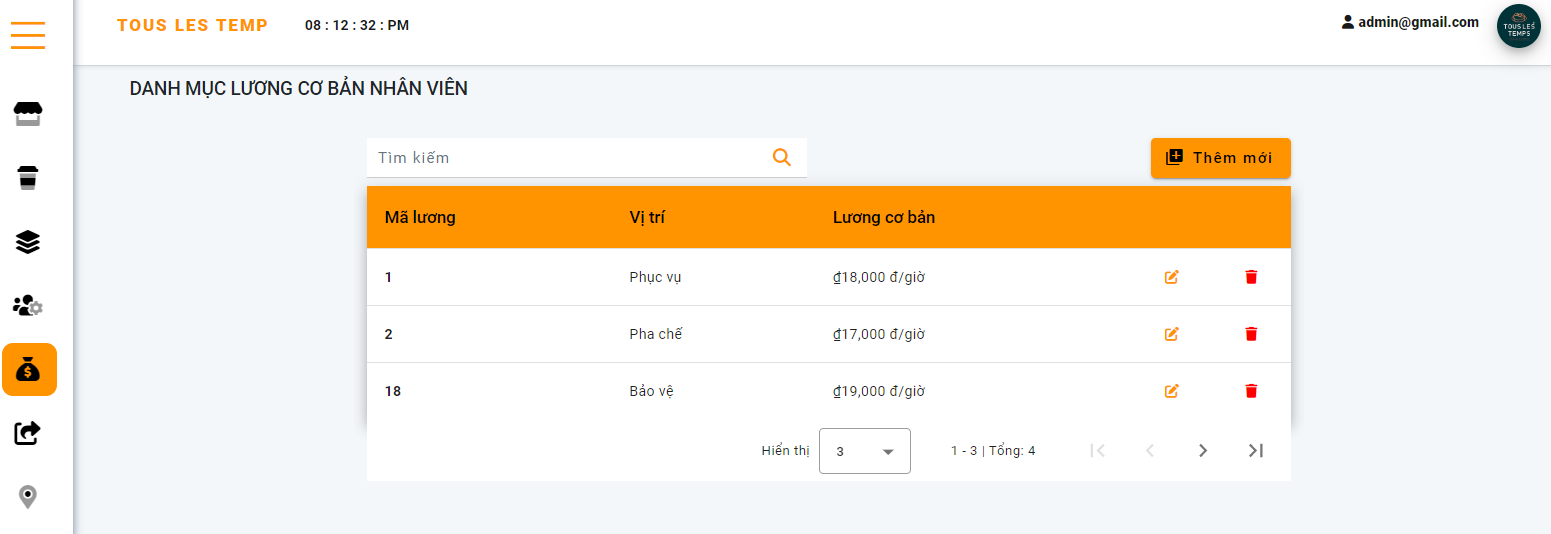


Hình 3. Danh sách nhóm nhân viên

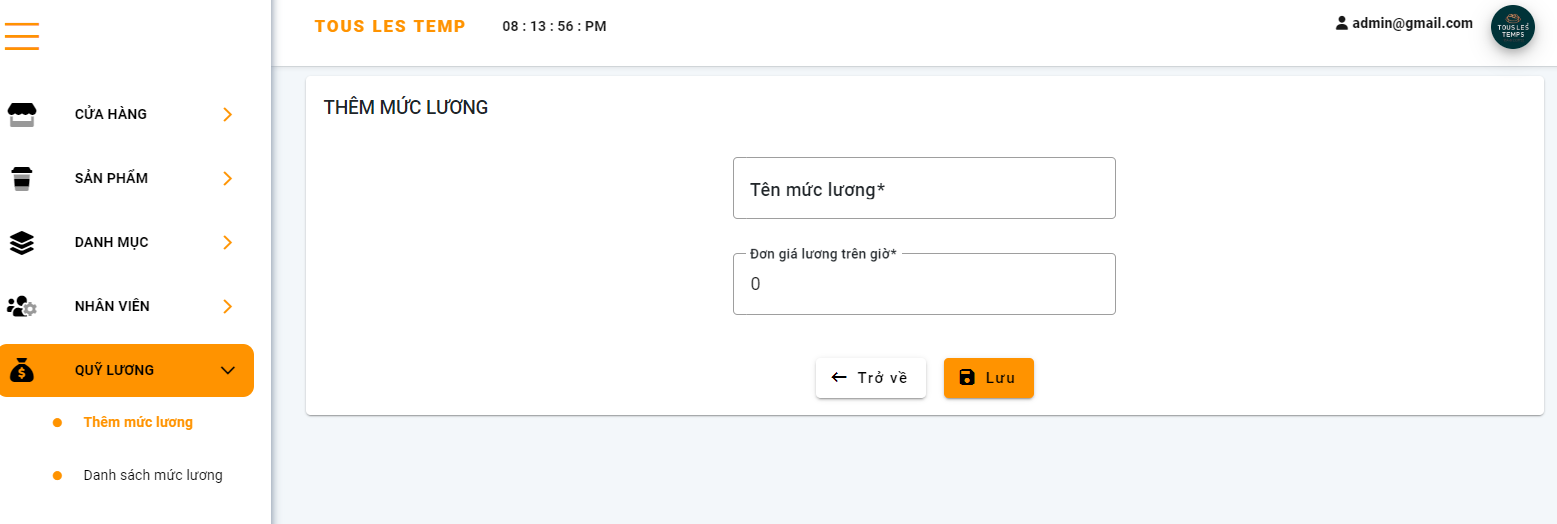
3.2.6 Giao diện quản lý chấm công

* Thông tin lương

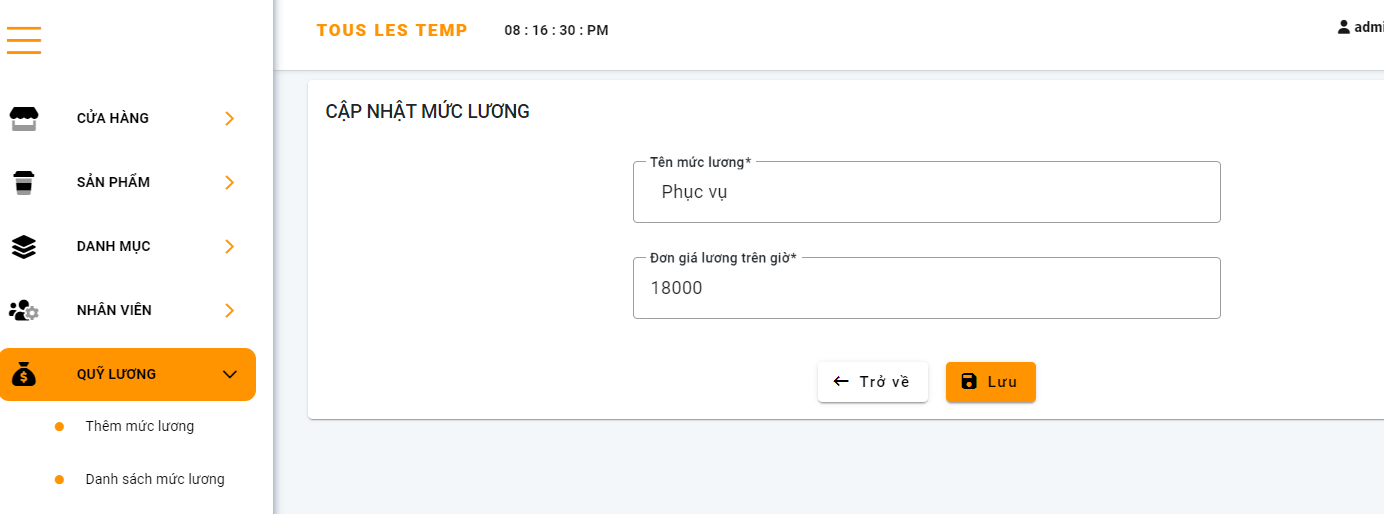
Thưc hiện chấm công nhân viên dựa trên số lượng ca làm việc, vị trí làm cũng như lương cơ bản từng loại vai trò. Hệ thống tạo chức năng này để người quản lý dễ dàng kiểm soát bảng chấm công cho từng nhân viên. Từ đó có thể phát triển và mở rộng, áp dụng cho mô hình lớn hơn và thực hiện chấm công kỹ lưỡng hơn. Thao tác đầu tiên là cập nhật thông tin lương cho cửa hàng, thêm mới mức lương



Hình 3. Danh sách lương cơ bản



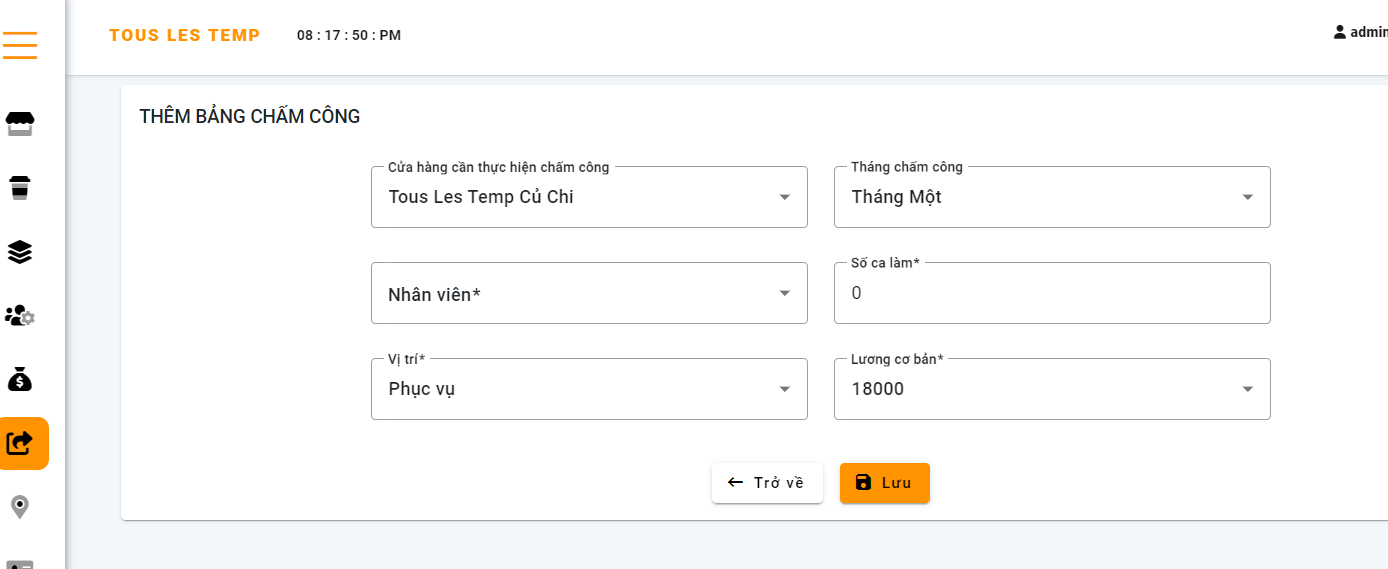
Hình 3. Tạo mới mức lương



Hình 3. Cập nhật thông tin lương

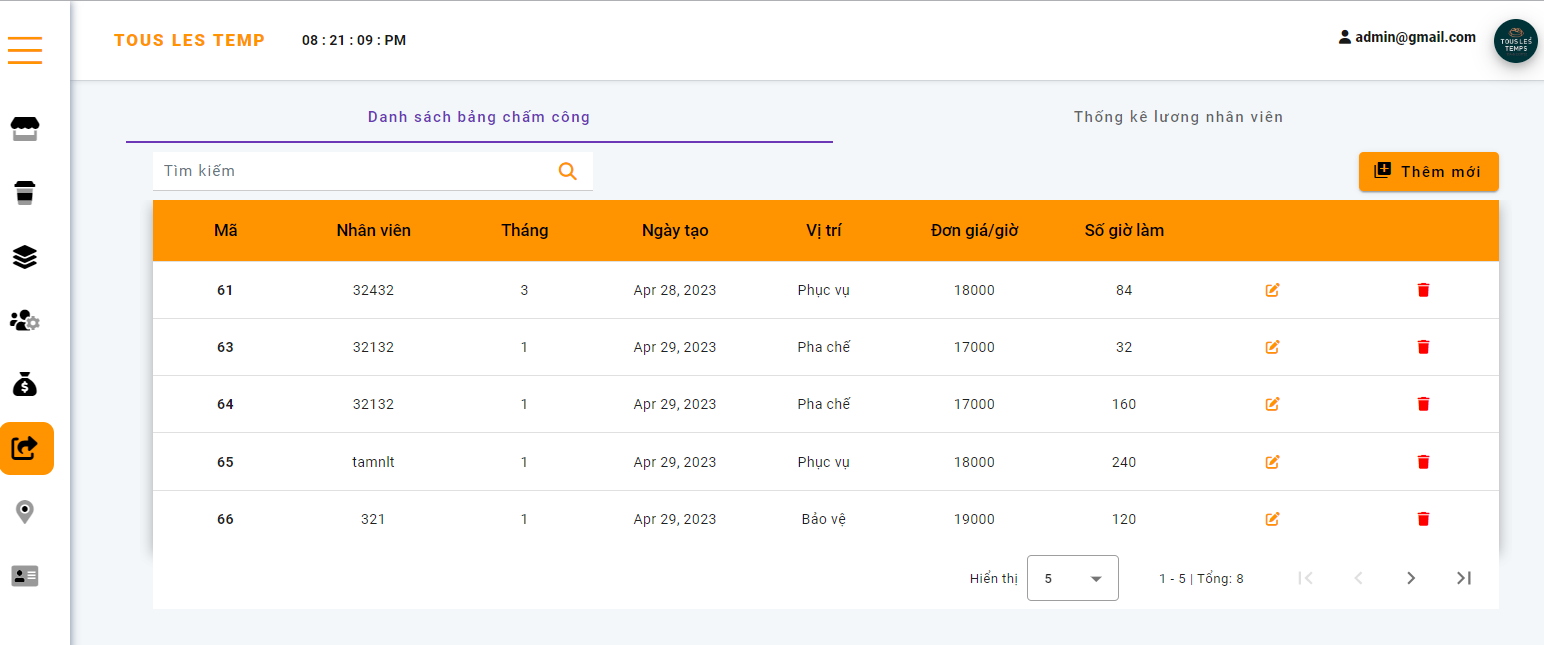
* Tạo mới bảng chấm công

Giao diện này cho phép người quản lý tiến hành tạo bảng chấm công cho nhân viên. Lựa chọn cửa hàng cần thực hiện sẽ hiển thi danh sách nhân viên của cửa hàng ấy, chọn tháng chấm công, cung cấp thông tin số giờ làm và vị trí làm

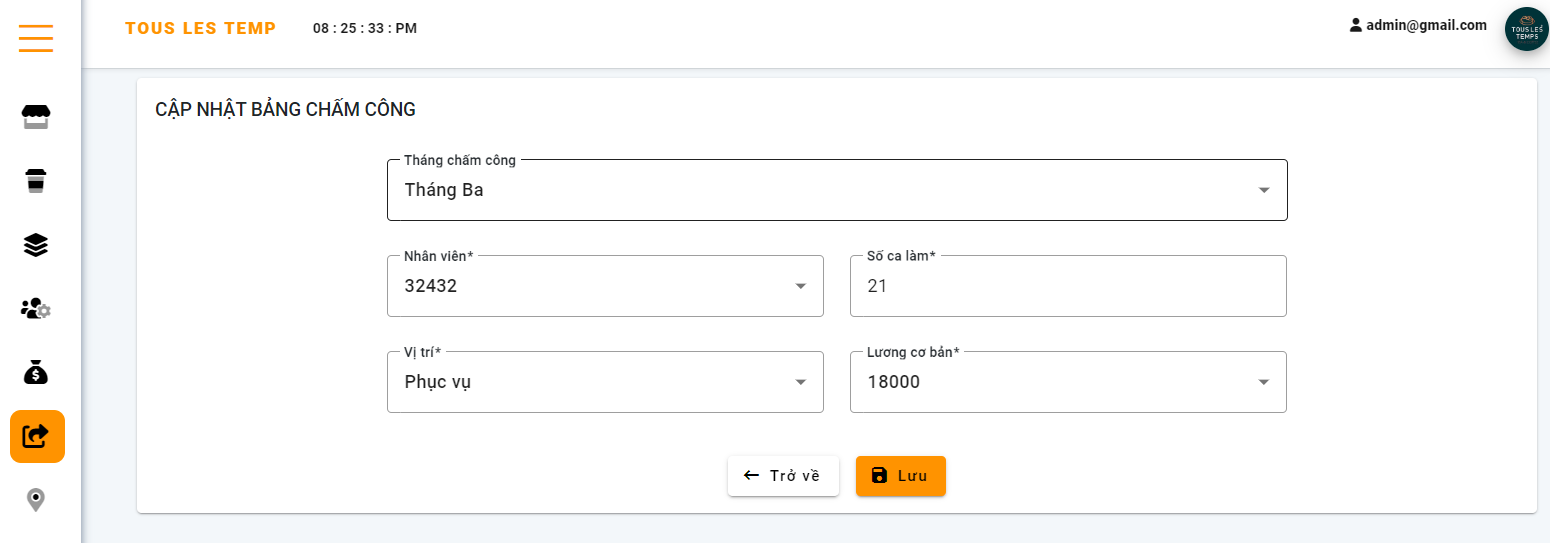


Hình 3. Tạo bảng chấm công

* Danh sách bảng chấm công

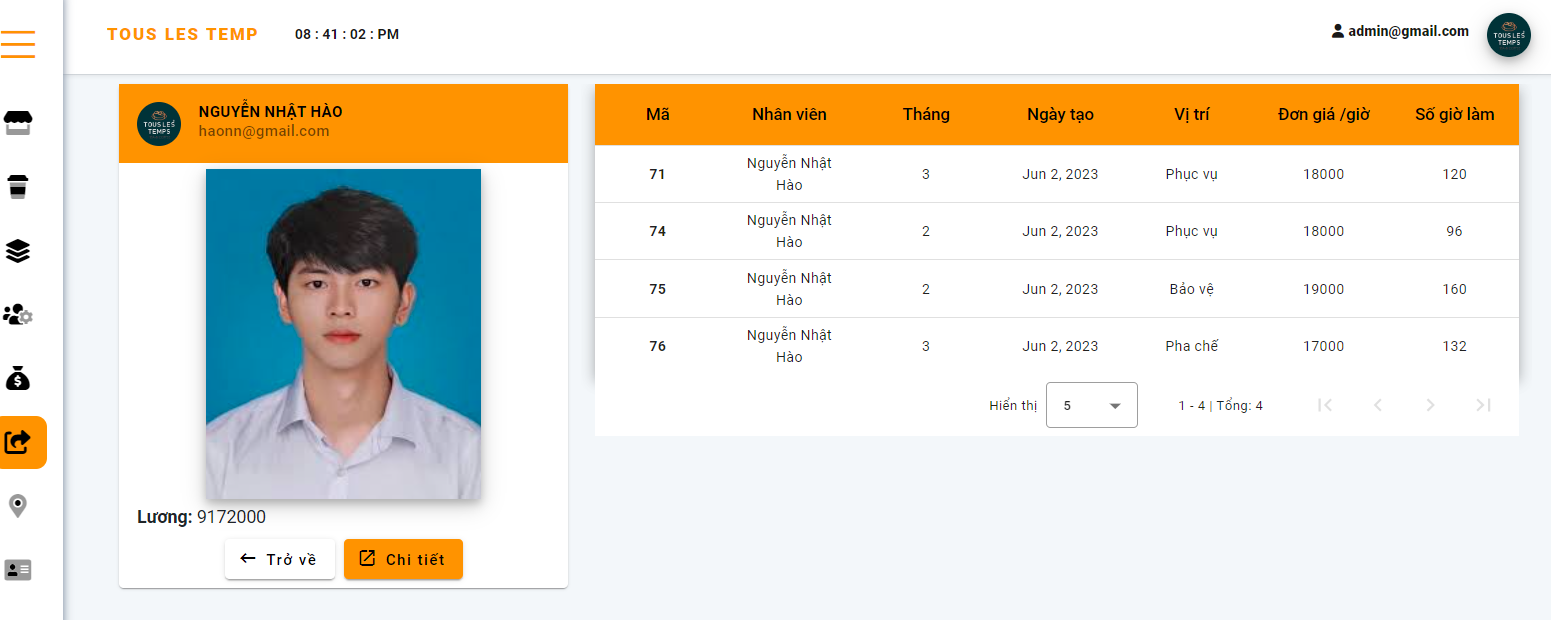


Hình 3. Quản lí danh sách bảng công

Ở màn hình này, ta có cái nhìn tổng quan về bảng chấm công của hệ thống, có thể thực hiện cập nhật xoá sửa cũng như phân trang cho danh sách bảng chấm công. Người quản lý sẽ dựa vào bảng chấm công này mà thực hiện tính lương cho nhân viên, đó nhận dữ liệu để thực hiện chức năng thông kê bằng biểu đồ cột

Hình 3. Cập nhật bảng chấm công

* Bảng công nhân viên



Hình 3. Bảng công nhân viên

Khi click chọn tên nhân viên trong danh sách bảng chấm công, ta sẽ tới giao diện bảng công chi tiết của nhân viên, bao gồm thông tin tổng lương nhân viên và 1 danh sách chi tiết bảng ca của nhân viên đó đã có trong hệ thống

* Thống kê lương nhân viên theo tháng

Hình 3. Thống kê lương nhân viên

Thông tin lương nhân viên được thống kê bằng Chart.js của Angular sẽ hiển thị ở đây, thực hiện chọn tháng cần thống kê, hệ thống sẽ thực hiện tính toán lương, chuyển đổi dữ liệu thu được ra dạng biểu đồ.

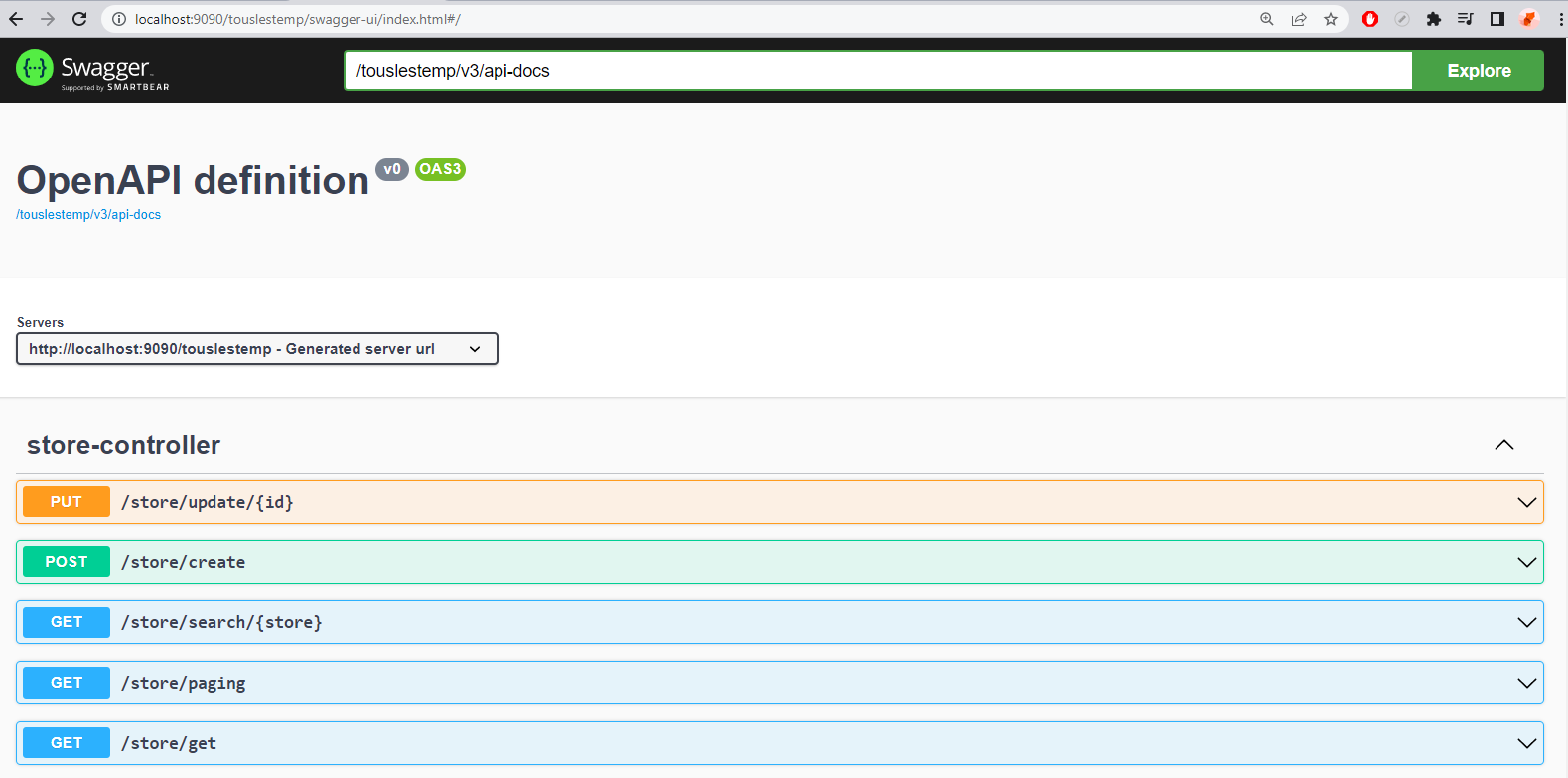
3.3 Một số xử lý tiêu biểu

3.3.1 Spring Boot

* Swagger Config

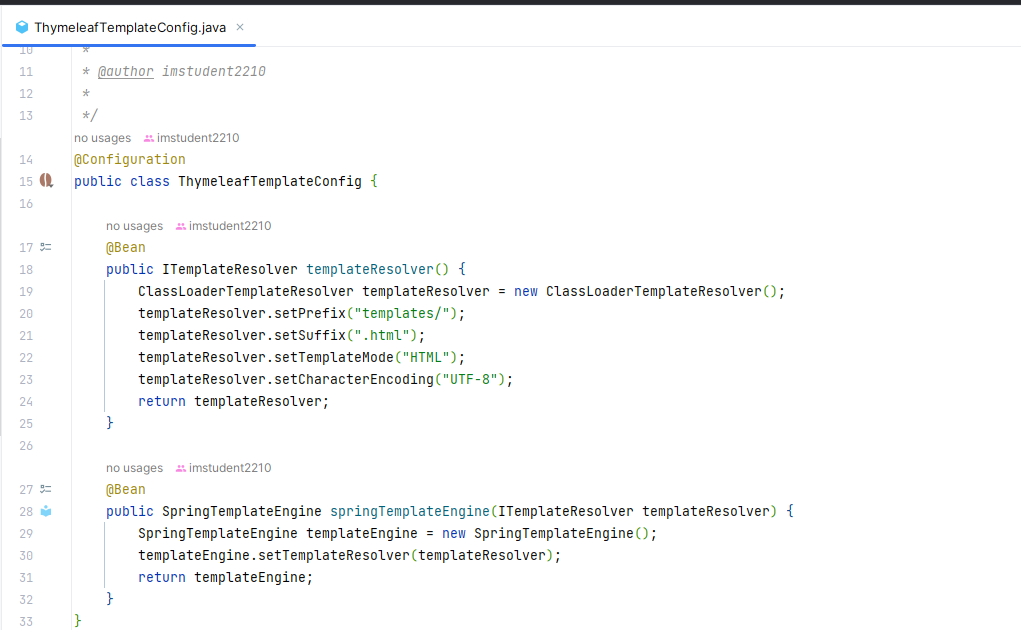


Hình 3. Cấu hình Swagger cho việc quản lý tài liệu RESTful API

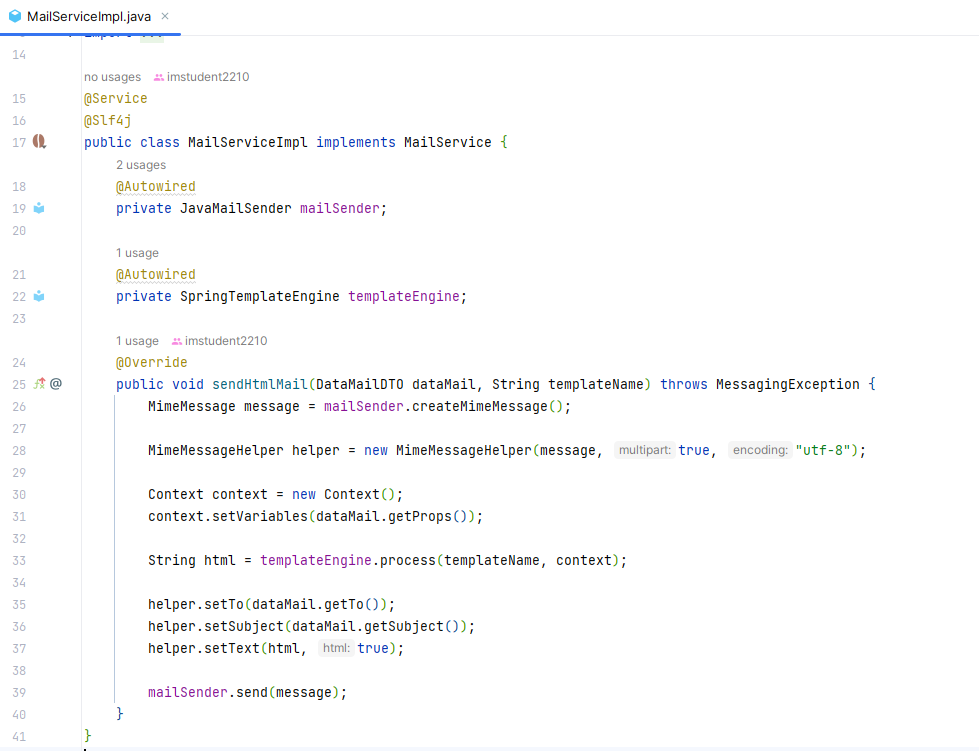


Hình 3. Giao diện Swagger

* Thymeleaf config - mail template

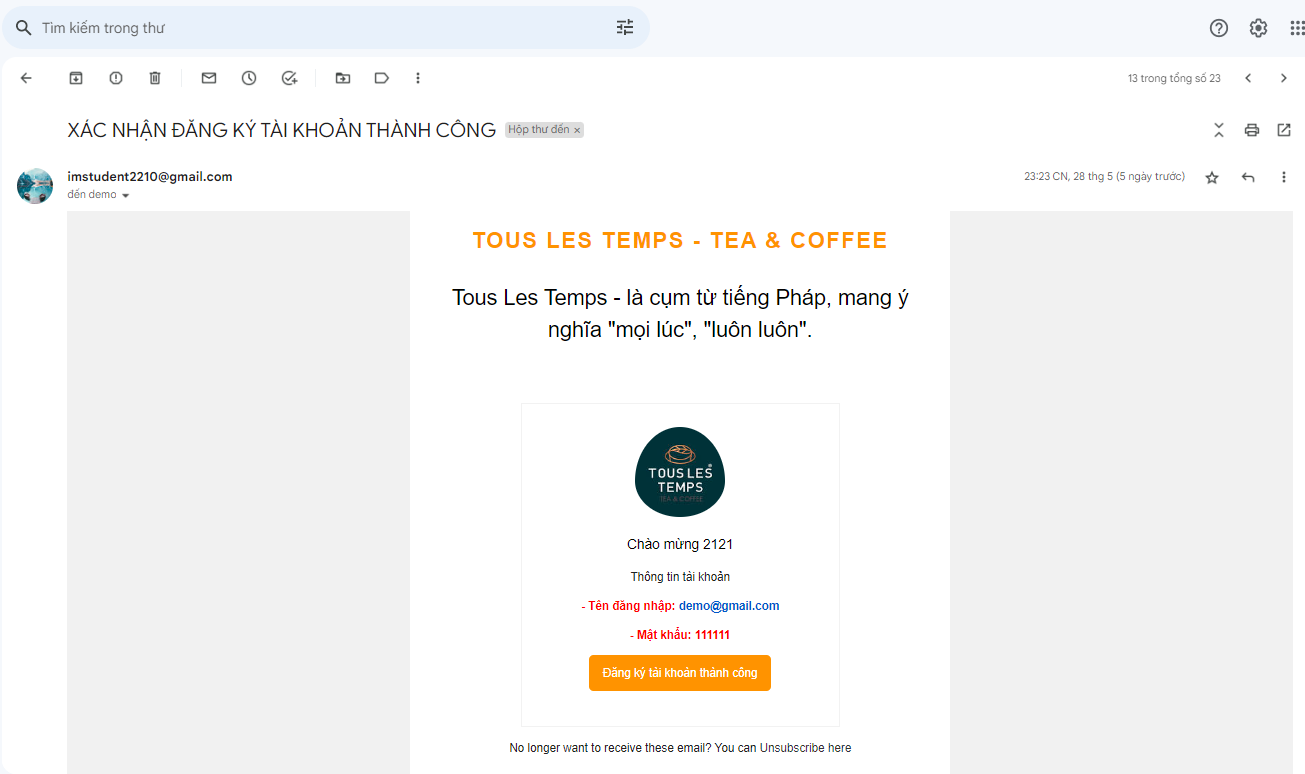


Hình 3. Cấu hình Thymeleaf cho mail template



Hình 3. Class MailServiceImpl

* Gửi mail với Thymeleaf



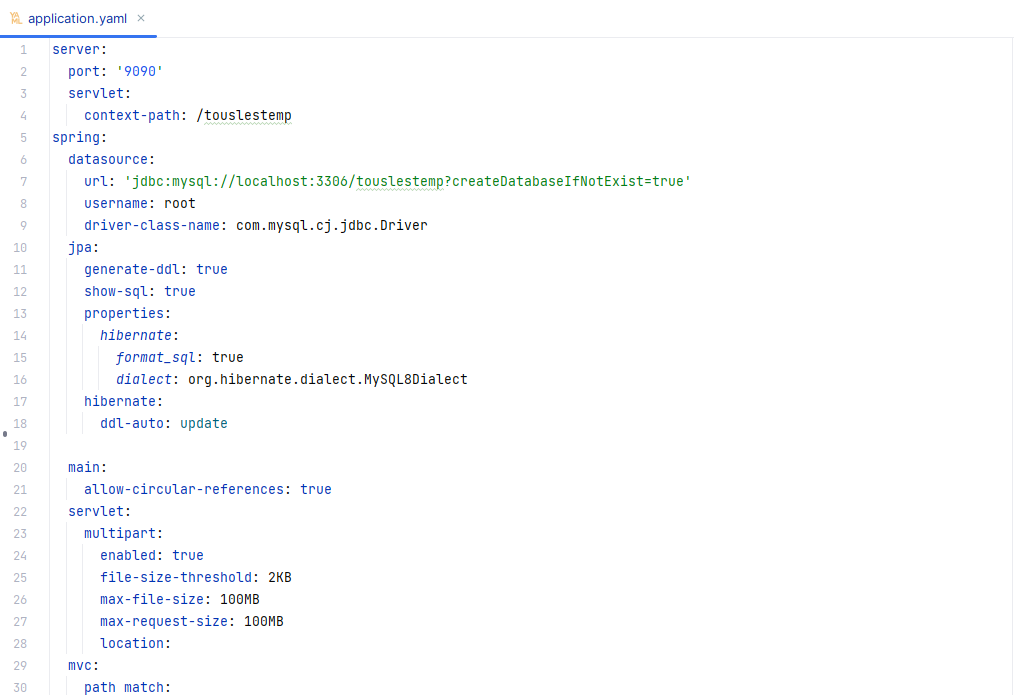
Hình 3. Mail xác thực đăng ký tài khoản thành công

* Security Config



Hình 3. Security Config

* Application.yaml file



Hình 3. File cấu hình của project application.yaml

3.3.2 Angular

* Interceptor

**Interceptor** được sử dụng để chèn mã xử lý trước và sau khi một yêu cầu HTTP được gửi đi hoặc nhận về. Trong trường hợp này, AuthInterceptor được sử dụng để thêm token xác thực vào tiêu đề (header) của yêu cầu HTTP và xử lý lỗi xác thực nếu cần.

Khai báo một lớp AuthInterceptor, được chú thích bằng @Injectable để đảm bảo rằng nó có thể được tiêm (inject) vào các thành phần khác trong ứng dụng.

* Lớp AuthInterceptor triển khai giao diện HttpInterceptor nó có một dependency injection của TokenStorageService trong constructor để có thể truy cập và quản lý token xác thực khi đăng nhập. Phương thức intercept() được triển khai từ giao diện HttpInterceptor và nhận vào hai tham số: req (yêu cầu HTTP đầu vào) và next (đối tượng xử lý yêu cầu tiếp theo trong chuỗi interceptor).
* Trong phương thức intercept(), tạo một bản sao của yêu cầu gốc bằng cách gán authReq = req, khai báo biến loginPath lưu đường dẫn '/login' và biến token lưu giá trị của token được lấy từ TokenStorageService.
* Kiểm tra nếu token khác null và không phải yêu cầu liên quan đến 'maps.google.com', khi đó tạo một bản sao của yêu cầu với tiêu đề (header) chứa token xác thực. Điều này được thực hiện bằng cách sử dụng phương thức clone() trên yêu cầu gốc và cung cấp một đối tượng cấu hình mới với tiêu đề mới được thiết lập bằng phương thức set().
* Cuối cùng, trả về kết quả của phương thức handle() trên đối tượng xử lý yêu cầu tiếp theo. Có thể áp dụng các xử lý khác nhau trên luồng dữ liệu được trả về bằng cách sử dụng các phương thức như pipe(), tap().
* Trong phần tap(), interceptor kiểm tra lỗi trả về từ yêu cầu HTTP. Nếu lỗi là một HttpErrorResponse và mã trạng thái không phải 401 (không được xác thực) hoặc đường dẫn hiện tại của trang không phải là '/login', thì không có hành động gì được thực hiện.



Hình 3. Class auth.interceptor.ts



Hình 3. Class home.guard.ts

Định nghĩa một guard (bảo vệ) trong Angular được sử dụng để kiểm tra xem người dùng có quyền truy cập vào một route cụ thể hay không. Trong trường hợp này, HomeGuard được sử dụng để bảo vệ route của trang chủ.

* Firebase Storage

Hình 3. Firebase config

Hàm onFileSelected() được sử dụng để đẩy file lên Firebase Storage, lấy URL tải xuống của file. Lưu trữ thông tin sản phẩm, nhân viên và cửa hàng lên firebase giúp cho cơ sở dữ liệu nhẹ nhàng trong việc truy xuất.

Ở project này, đã sử dụng Firebase Realtime Database, là một cơ sở dữ liệu thời gian thực (real-time) lưu trữ dữ liệu dạng cây JSON. Nó cho phép lưu giữ và đồng bộ hoá dữ liệu trực tiếp giữa ứng dụng khác nhau mà không cần làm việc với các REST API truyền thống. Firebase Realtime Database rất hữu ích trong việc xây dựng các ứng dụng cần cập nhật dữ liệu trực tiếp và chia sẻ dữ liệu trực tuyến giữa nhiều người dùng.

KẾT LUẬN

* **Kết quả đạt được**
* Thực hiện được ứng dụng sử dụng Spring Framwork và tích hợp Angular cho giao diện người dùng, bảo mật hệ thống chặt chẽ, từ bảo mật truy cập từng API với security config kết hợp JWT token để bảo mật API. Thực hiện bảo mật đăng nhập hệ thống với quy trình xác thực hai yếu tố 2FA.
* Thực hiện cấu hình và sử dụng mail sender cho ứng dụng, thực hiện đăng ký thành công, sẽ có mail xác nhận thành công được gửi ngay lập tức
* Thông kê bằng biểu đồ về lương nhân viên, số lượng nhân viên trong từng nhóm nhân viên, tạo được biểu đồ với việc lấy dữ liệu chính xác từ RESTful API của backend, giao diện biểu đồ rõ ràng, cụ thể trực quan, giúp người quản lý dễ dàng nhìn nhận đánh giá
* Tìm hiểu cũng như thực hiện các công nghệ như: JPA, Hibernate trong Spring để làm việc với cơ sở dữ liệu quan hệ MySQL
* Tích hợp Swagger cho việc cung cấp một cách đơn giản để mô tả và quản lý các endpoint của API, bao gồm các tham số, phương thức, tài liệu, và các ví dụ sử dụng. Swagger tự động tạo tài liệu cho API dựa trên các mô tả và đánh dấu trong mã nguồn của dự án. Nó tạo ra một giao diện tài liệu web tương tác, giúp ta thử nghiệm các endpoint và phương thức của API nhanh chóng
* Sử dụng công nghệ Google Map API để quản lí địa chỉ các cửa hàng, giúp cho việc tích hợp vào giao diện của người dùng sau này để tìm kiếm đến các cửa hàng nhanh chóng, chính xác
* Thiết kế giao diện chương trình dễ sử dụng, sinh động
* Các tính năng của chương trình đã hoàn thiện:
* Quản lý danh sách cửa hàng
* Quản lý sản phẩm.
* Quản lý nhân viên.
* Thực hiện chấm công
* Báo cáo thống kê.
* **Hạn chế**
* Chưa triển khai việc quản lý tấc cả sản phẩm cùng thuộc nhiều cửa hàng, chưa thực hiện tính năng mua hàng của người dùng.
* Chực thực hiện quản lý số lượng sản phẩm rõ ràng, cũng như nguyên liệu cho sản phẩm
* Chưa thực hiện kiểm soát tên các cửa hàng giống nhau chính xác
* **Kết luận**

Kết quả nghiên cứu đã áp dụng Spring và Spring Security cho ứng dụng, thực hiện đúng các giai đoạn của phát triển phần mềm. Việc áp dụng công nghệ nêu trên vào việc quản lý hệ thống chuỗi cửa hàng Tous Les Temp như vậy là hợp lý, đáp ứng đủ các chức năng cơ bản ở phía quản trị viên hệ thống.

* **Hướng phát triển**
* Thực hiện chấm công nhân viên chính xác hơn thông qua giờ vào ca, giờ ra ca
* Kiểm soát thêm các nguồn thông tin như: nguyên liệu, công thức, cơ sở vật chất từng cửa hàng của hệ thống
* Xây dựng thêm giao diện cho người dùng có thể tìm sản phẩm và đặt mua thông qua gọi số điện thoại, giúp người dùng có một trang web hoàn thiện hơn, tích hợp Google Map để chỉ đường đến của hàng.

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | H. T. T. Ngà, in *Phân tích thiết kế hệ thống thông tin, Trường đại học Nha Trang*, 2019. |
| [2] | N. T. Đ. Trang, in *Bài giảng hệ thống thông tin quản lý, Trường đại học Nha Trang*. |
| [3] | JetBrain, "IntelliJ IDEA," [Online]. Available: https://www.jetbrains.com/idea/. [Accessed 16 05 2023]. |
| [4] | V. Tanzu, "Why Spring?," [Online]. Available: https://spring.io/why-spring. [Accessed 16 05 2023]. |
| [5] | "Microsoft Visual Studio Code," [Online]. Available: https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html. [Accessed 16 05 2023]. |
| [6] | Mkyong, "Spring Boot Tutorials," [Online]. Available: https://mkyong.com/tutorials/spring-boot-tutorials/. [Accessed 01 03 2023]. |
| [7] | P. T. T. Thuý, in *Bài giảng Hệ quản trị cơ sở dữ liệu, Trường đại học Nha Trang*, 2019. |
| [8] | C. Walls, in *Spring Boot in Action*. |